NT04174US

#3

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年 ２月２２日

出願番号
Application Number: 特願２００１－０４６２５０

出願人
Applicant(s): 株式会社日立製作所

U.S. Appln. Filed 8-29-01
Inventor: T. Endo et al
Mattingly Stanger & Malur
Docket NIT295

CERTIFIED COPY OF PRIORITY DOCUMENT

２００１年 ７月２７日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

出証番号 出証特２００１－３０６６０２６

【書類名】　特許願
【整理番号】　NT00P0641
【提出日】　平成13年 2月22日
【あて先】　特許庁長官　殿
【国際特許分類】　G06K 19/073
【発明者】
【住所又は居所】　東京都国分寺市東恋ヶ窪一丁目２８０番地　株式会社日立製作所　中央研究所内
【氏名】　遠藤　隆
【発明者】
【住所又は居所】　東京都国分寺市東恋ヶ窪一丁目２８０番地　株式会社日立製作所　中央研究所内
【氏名】　神永　正博
【発明者】
【住所又は居所】　東京都国分寺市東恋ヶ窪一丁目２８０番地　株式会社日立製作所　中央研究所内
【氏名】　渡邊　高志
【発明者】
【住所又は居所】　東京都国分寺市東恋ヶ窪一丁目２８０番地　株式会社日立製作所　中央研究所内
【氏名】　大木　優
【特許出願人】
【識別番号】　000005108
【氏名又は名称】　株式会社日立製作所
【代理人】
【識別番号】　100068504
【弁理士】
【氏名又は名称】　小川　勝男
【電話番号】　03-3661-0071

【選任した代理人】

【識別番号】　100086656

【弁理士】

【氏名又は名称】　田中　恭助

【電話番号】　03-3661-0071

【選任した代理人】

【識別番号】　100094352

【弁理士】

【氏名又は名称】　佐々木　孝

【電話番号】　03-3661-0071

【手数料の表示】

【予納台帳番号】　081423

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】明細書

【発明の名称】　情報処理装置

【特許請求の範囲】

【請求項１】　プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行する中央演算装置とを有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、

一つのデータ処理手段が、その入力データを処理し、処理済データを出力する入力データ処理手段を含み、且つ

生成される撹乱用データXiのハミングウエイトが常に一定値であり、且つ、生成されたXiを前記入力データ処理手段で処理済撹乱用データXoに変換した後のハミングウエイトも常に一定になるような、撹乱用データXiを生成する撹乱用データ生成手段を有し、

前記撹乱用データ生成手段は、

前記撹乱用データXiを使って入力データD1を変形し、変形データH1を作成するデータ変形手段と、

変形データH1を前記入力データ処理手段と同じ処理を行い、処理済変形データH2を得る変形データ処理手段と、

前記撹乱用データXiに前記入力データ処理手段と同じ処理を行い、処理済撹乱用データXoを生成する撹乱用データ処理手段と、

処理済撹乱用データXoを使用して処理済変形データH2を処理し、

入力データD1を入力データ処理手段で処理した結果である処理済データD2を得るデータ逆変形処理手段と、を有することを特徴とする情報処理装置。

【請求項２】　生成される撹乱用データXiのハミングウエイトが、常に一定値であり、且つ、生成されたXiを予め定義された入力データ処理に従って処理して生成した処理済撹乱用データXoのハミングウエイトも常に一定であるような、撹乱用データおよび処理済撹乱用データ生成手段であって、

撹乱用データおよび処理済撹乱用データを生成する手段は、ハミングウエイト

一定の乱数を生成する撹乱用データ生成手段と、撹乱用データ処理手段と、及びハミングウエイト検査手段とを有し、

前記撹乱用データ生成手段で生成されたデータを撹乱用データ処理手段で処理し、処理結果をハミングウエイト検査手段でハミングウエイトが所定の値であるかを検査し、所定の値でない場合に、撹乱用データ生成手段のデータ生成からやり直すように制御信号を送るハミングウエイト検査手段を有し、ハミングウエイトが一定で、撹乱用データ処理手段で処理した後も、ハミングウエイトが一定となるような撹乱用データを生成するように構成された、撹乱用データおよび処理済撹乱用データ生成手段を、有する情報処理装置。

【請求項３】　前記撹乱用データおよび処理済撹乱用データを生成する手段が、あらかじめ、ハミングウエイトが一定でかつ、撹乱用データ処理手段で処理を行った結果も、ハミングウエイトが一定となるようなデータを選択して複数格納した撹乱用データ格納手段と、撹乱用データ格納手段に格納されたデータをランダムに選択する撹乱用データ選択手段と、選択された撹乱用データXiを処理して、処理済の撹乱用データを生成する撹乱用データ処理をなす手段を有することを特徴とした、請求項１に記載の情報処理装置。

【請求項４】　前記撹乱用データおよび処理済撹乱用データを生成する手段として、あらかじめ、ハミングウエイトが一定でかつ、撹乱用データ処理手段で処理を行った結果もハミングウエイトが一定となるようなデータを、撹乱用データ処理手段で処理した結果と組み合わせて、複数組格納した撹乱用データおよび処理済撹乱用データ格納手段と、撹乱用データおよび処理済撹乱用データ格納手段に格納されたデータをランダムに選択する撹乱用データおよび処理済撹乱用データ選択手段とを有することを特徴とした、請求項１に記載の情報処理装置。

【請求項５】　前記撹乱用データおよび処理済撹乱用データ格納手段に格納するデータの組の数が遇数個であり、且つ撹乱用データおよび処理済撹乱用データのいずれのビットについても、０となるか１となるかの確率が0.5となるように選択したデータを格納している撹乱用データおよび処理済撹乱用データ格納手段を有することを特徴とする、請求項４に記載の情報処理装置。

【請求項６】　ハミングウエイト一定の乱数を生成する手段として、発生する乱

数のビット数の半分のビット数の乱数を発生させる乱数発生手段と、ビット反転演算を計算するビット反転演算手段と、乱数発生手段で生成されたデータと、ビット反転演算手段の計算結果を結合して、所定のビット数の数値を生成する、データ結合手段から成る、ハミングウエイト一定乱数生成手段を、有することを特徴とする情報処理装置。

【請求項７】　ハミングウエイト一定の乱数を生成する手段として、乱数発生手段と、前記乱数発生手段により生成された乱数のハミングウエイト計算手段と、前記ハミングウエイト計算手段の結果を検査し、目標ハミングウエイトと等しくない場合に乱数発生手段に対して乱数の再生成を行わせる、ハミングウエイト検査手段とを有することを特徴とした、ハミングウエイト一定乱数生成手段を、有することを特徴とする情報処理装置。

【請求項８】　ハミングウエイト一定の乱数を生成する手段として、発生させたい乱数のビット数の約数となるビット数ハミングウエイト一定の乱数を発生させるハミングウエイト一定部分乱数発生手段と、発生させたい乱数のビット数に達するまで、前記ハミングウエイト一定部分乱数発生手段に乱数を生成させる乱数発生制御手段と、発生した乱数を結合して、所定のビット数の乱数とするデータ結合手段からなる、ハミングウエイト一定乱数生成手段を、有することを特徴とする情報処理装置。

【請求項９】　プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行する中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段を有し、

一つのデータ処理手段が、インデックスとそれに対応したデータから成る表を有し、入力データをインデックスとして表引きを行い、インデックスに対応したデータを処理済データとして出力する入力データ処理手段を有し、

ハミングウエイトの値が常に一定となる第１の撹乱用データX1iと、

ハミングウエイトの値が常に一定となる第２の撹乱用データX2iと、

第１撹乱用データX1iでインデックスを撹乱し、

第２撹乱用データX2iでインデックスに対応したデータを撹乱することにより生成された変形済み表を使い、

前記撹乱用データXiを使って入力データD1を変形し、変形データH1を作成するデータ変形手段と、

変形データH1をインデックスとして、前記変形済み表を引いて変形データＨ２を取り出す変形済み表アクセス処理手段と、

変形データＨ２を第２撹乱用データX2iを用いて、

入力データD1を入力データ処理手段で処理した結果である処理済データD2を得るデータ逆変形処理手段とを有する、ことを特徴とする情報処理装置。

【請求項１０】　第１撹乱用データX1iおよび第２撹乱用データX2iおよび変形済み表を生成する手段として、第１撹乱用データX1iを生成するための第１ハミングウエイト一定乱数生成手段と、第２撹乱用データX2iを生成するための第２ハミングウエイト一定乱数生成手段と、第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段を有すること、を特徴とする請求項９に記載の情報処理装置。

【請求項１１】　第１撹乱用データX1iおよび第２撹乱用データX2iおよび変形済み表を生成する手段として、ハミングウエイト一定の値を予め複数格納している第１撹乱用データ格納手段と、

第１撹乱用データ格納手段に格納されたデータをランダムに選択し、第１撹乱用データX1iとするための第１撹乱用データ選択手段と、

ハミングウエイト一定の値を予め複数格納している第２撹乱用データ格納手段と、第２撹乱用データ格納手段に格納されたデータをランダムに選択し、第２撹乱用データX2iとするための第２撹乱用データ選択手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段を有することを特徴とする、請求項９に記載の情報処理装置。

【請求項１２】　第１撹乱用データX1iおよび第２撹乱用データX2iおよび変形済

み表を生成する手段として、第１撹乱用データになりうる、ハミングウエイト一定の値と、第２撹乱用データになりうる、ハミングウエイト一定の値と前記第１撹乱用データとなり得る値と前記第２撹乱用データとなり得る値と表を用いて、あらかじめ表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形して作成した変形済み表を、作成したときに用いた第１撹乱用データと第２撹乱用データを組にしたものを、複数組格納した第一撹乱用データ、第２撹乱用データおよび変形済み表格納手段と、

第一撹乱用データ、第２撹乱用データおよび変形済み表格納手段からランダムに、

第１撹乱用データと第２撹乱用データと変形済み表を選択する、第１撹乱用データ第２撹乱用データおよび変形済み表選択手段を有することを特徴とする、請求項９に記載の情報処理装置。

【請求項１３】　プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行する中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、

一つのデータ処理手段が、表引きを行い、表引き結果をデータ処理処理し、処理結果を処理済データとして出力する入力データ処理手段を有し、且つ、

ハミングウエイトの値が常に一定となる第１の撹乱用データX1iと、

ハミングウエイトの値が常に一定となり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウエイトも常に一定となる様な第２の撹乱用データX2iと、前記第２の撹乱用データX2iをデータ処理した結果の処理済第２撹乱用データX2iと、

第１撹乱用データX1iでインデックスを撹乱し、

第２撹乱用データX2iでインデックスに対応したデータを撹乱することにより生成された変形済み表を使い、

前記撹乱用データXiを使って入力データDiを変形し、変形データHiを作成するデータ変形手段と、

変形データH1をインデックスとして、前記変形済み表を引いて変形データＨ２を取り出す変形済み表アクセス処理手段と、

処理済変形データＨ２を処理し、処理済変形データＨ３を得る変形済みデータ処理手段と、

処理済み変形データＨ３を処理済第２撹乱用データX2oを用いて、

入力データD1を用いて表引きを行い、表引き結果をデータ処理処理した結果である処理済データD2を得るデータ逆変形処理手段とを有することを特徴とする情報処理装置。

【請求項１４】　第１撹乱用データX1iおよび第２撹乱用データX2iおよび処理済第２撹乱用データX2oおよび変形済み表を生成する手段として、

第１撹乱用データX1iを生成するための第１ハミングウエイト一定乱数生成手段と、

第２撹乱用データX2iを生成するための第２ハミングウエイト一定乱数生成手段と、

第２撹乱用データX2iを処理し、処理済第２撹乱用データX2oを生成する撹乱用データ処理手段と、

処理済第２撹乱用データX2oのハミングウエイトを計算し、処理済第２撹乱用データX2oのハミングウエイトが不適切な場合は第２ハミングウエイト一定乱数生成手段に対して、

第２撹乱用データX2iの再生成を行わせる、ハミングウエイト検査手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項１３に記載の情報処理装置。

【請求項１５】　前記第１撹乱用データX1iおよび第２撹乱用データX2iおよび処理済第２撹乱用データX2oおよび変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第１撹乱用データ格納手段と、

第１撹乱用データ格納手段に格納されたデータをランダムに選択し、第１撹乱

用データX1iとするための第１撹乱用データ選択手段と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な値を複数格納している第２撹乱用データ格納手段と、第２撹乱用データ格納手段に格納されたデータをランダムに選択し、第２撹乱用データX2iとするための第２撹乱用データ選択手段と、

第２撹乱用データX2iを処理し、処理済第２撹乱用データX2oを生成する撹乱用データ処理手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項１３に記載の情報処理装置。

【請求項１６】　前記第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第１撹乱用データ格納手段と、

第１撹乱用データ格納手段に格納されたデータをランダムに選択し、第１撹乱用データX1iとするための第１撹乱用データ選択手段と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第２撹乱用データと処理済第２撹乱用データの組を複数組格納している第２撹乱用データおよび処理済第２撹乱用データ格納手段と、第２撹乱用データおよび処理済第２撹乱用データ格納手段に格納された、第２撹乱用データと処理済第２撹乱用データの組をランダムに選択し、第２撹乱用データX2iおよび処理済第２撹乱用データX2oとするための第２撹乱用データおよび処理済第２撹乱用データ選択手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段と、を有することを特徴とする、請求項１３に記載の情報処理装置。

【請求項１７】　前記第１撹乱用データX1iおよび処理済第２撹乱用データX2oお

よび変形済み表を生成する手段として、

ハミングウエイト一定の第１撹乱用データに使用できる値と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第２撹乱用データに使用できる値を処理した結果選られる処理済第２撹乱用データに使用できる値と、

第１撹乱用データに使用できる値と前記処理済第２撹乱用データに使用できる値を生成したときに用いた第２撹乱用データに使用できる値を用いて、表のインデックスを第１撹乱用データX1iに使用できる値で変形し、表のデータを第２撹乱用データX2iに使用できる値で変形し、変形済み表を生成し、

前記第１撹乱用データに使用できる値と処理済第２撹乱用データに使用できる値と変形済み表の組を複数格納した、第１撹乱用データおよび処理済第２撹乱用データおよび変形済み表格納手段と、

第１撹乱用データと処理済第２撹乱用データと変形済み表の組をランダムに選択し、

第１撹乱用データX1iと処理済第２撹乱用データX2oと処理済表とするための、第１撹乱用データおよび処理済第２撹乱用データおよび変形済み表選択手段とを有することを特徴とする、請求項１３に記載の情報処理装置。

【請求項１８】　プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行し、データ処理を行なう中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段を有し、

一つのデータ処理手段が、表引きを行い、表引き結果をデータ処理処理し、処理結果を処理済データとして出力する入力データ処理手段を含み、前記処理手段を複数回繰り返すことで処理結果を生成し、且つ、

ハミングウエイトの値が常に一定となる第１の撹乱用データX1iと、

ハミングウエイトの値が常に一定となり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウエイトも常に一定となる様な第２の撹乱用データX2

iと、前記第２の撹乱用データX2iをデータ処理した結果の処理済第２撹乱用データX2oと、第１撹乱用データX1iでインデックスを撹乱し、第２撹乱用データX2iでインデックスに対応したデータを撹乱することにより生成された変形済み表を使い、

前記撹乱用データXiを使って入力データD1を変形し、変形データH1を作成するデータ変形手段と、

変形データH1をインデックスとして、前記変形済み表を引いて変形データＨ２を取り出す変形済み表アクセス処理手段と、

処理済変形データＨ２を処理し、処理済変形データＨ３を得る変形済みデータ処理手段と、

第１撹乱用データX1iを用いて処理済み変形データＨ３を変形し、処理済変形データＨ４を得るデータ変形処理手段と、

処理済第２撹乱用データX2oを用いて処理済み変形データＨ４を変形し、処理済変形データＨ５を得るデータ変形処理手段と、

処理済変形データＨ５をインデックスとして、前記変形済み表を引いて変形データＨ６を取り出す変形済み表アクセス処理手段と、

処理済変形データＨ６を処理し、処理済変形データＨ７を得る変形済みデータ処理手段と、

処理済第２撹乱用データX2oを用いて処理済み変形データＨ７を変形し、入力データD1を用いて表引きを行い、表引き結果をデータ処理処理し、さらにその結果を用いて表引きを行い、表引き結果をデータ処理処理した結果である処理済データD2を得るデータ逆変形処理手段と、を有することを特徴とする情報処理装置。

【請求項１９】　第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび変形済み表を生成する手段として、

第１撹乱用データX1iを生成するための第１ハミングウエイト一定乱数生成手段と、

第２撹乱用データX2iを生成するための第２ハミングウエイト一定乱数生成手段と、

第２撹乱用データX2iを処理し、処理済第２撹乱用データX2oを生成する撹乱用データ処理手段と、

処理済第２撹乱用データX2oのハミングウエイトを計算し、処理済第２撹乱用データX2oのハミングウエイトが不適切な場合は第２ハミングウエイト一定乱数生成手段に対して、

第２撹乱用データX2iの再生成を行わせる、ハミングウエイト検査手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項１８に記載の情報処理装置。

【請求項２０】　前記第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第１撹乱用データ格納手段と、

第１撹乱用データ格納手段に格納されたデータをランダムに選択し、第１撹乱用データX1iとするための第１撹乱用データ選択手段と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な値を複数格納している第２撹乱用データ格納手段と、第２撹乱用データ格納手段に格納されたデータをランダムに選択し、第２撹乱用データX2iとするための第２撹乱用データ選択手段と、

第２撹乱用データX2iを処理し、処理済第２撹乱用データX2oを生成する撹乱用データ処理手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項１８に記載の情報処理装置。

【請求項２１】　第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第１撹乱用データ格納手段と

第１撹乱用データ格納手段に格納されたデータをランダムに選択し、第１撹乱用データX1iとするための第１撹乱用データ選択手段と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第２撹乱用データと処理済第２撹乱用データの組を複数組格納している第２撹乱用データおよび処理済第２撹乱用データ格納手段と、第２撹乱用データおよび処理済第２撹乱用データ格納手段に格納された、第２撹乱用データと処理済第２撹乱用データの組をランダムに選択し、第２撹乱用データX2iおよび処理済第２撹乱用データX2oとするための第２撹乱用データおよび処理済第２撹乱用データ選択手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段とを有することを特徴とする、請求項１８に記載の情報処理装置。

【請求項２２】　第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび変形済み表を生成する手段として、

ハミングウエイト一定の第１撹乱用データに使用できる値と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第２撹乱用データに使用できる値を処理した結果選られる処理済第２撹乱用データに使用できる値と、

第１撹乱用データに使用できる値と前記処理済第２撹乱用データに使用できる値を生成したときに用いた第２撹乱用データに使用できる値を用いて、表のインデックスを第１撹乱用データX1iに使用できる値で変形し、表のデータを第２撹乱用データX2iに使用できる値で変形し、変形済み表を生成し、

前記第１撹乱用データに使用できる値と処理済第２撹乱用データに使用できる値と変形済み表の組を複数格納した、第１撹乱用データおよび処理済第２撹乱用データおよび変形済み表格納手段と、

第１撹乱用データと処理済第２撹乱用データと変形済み表の組をランダムに選択し、第１撹乱用データX1iと処理済第２撹乱用データX2oと処理済表とするため

の、第１撹乱用データおよび処理済第２撹乱用データおよび変形済み表選択手段とを有することを特徴とする、請求項１８に記載の情報処理装置。

【請求項２３】　プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行し、データ処理を行なう中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、

一つのデータ処理手段が、表引きを行い、表引き結果をデータ処理処理し、処理結果を処理済データとして出力する入力データ処理手段を含み、前記処理手段を複数回繰り返すことで処理結果を生成し、且つ、

ハミングウエイトの値が常に一定である第１の撹乱用データX1iと、

ハミングウエイトの値が常に一定であり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウエイトも常に一定となる様な第２の撹乱用データX2iと、前記第２の撹乱用データX2iをデータ処理した結果の処理済第２撹乱用データX2oと、

ハミングウエイトの値が常に一定である第３の撹乱用データX3iと、

ハミングウエイトの値が常に一定であり、かつ前記表引き後に行われるデータ処理を行った結果のハミングウエイトも常に一定となる様な第４の撹乱用データX4iと、前記第４の撹乱用データX4iをデータ処理した結果の処理済第４撹乱用データX4oと、

第１撹乱用データX1iを用いてインデックスを撹乱し、

さらに第３撹乱用データX3iで前記第一撹乱用データを用いた撹乱されたインデックスをさらに撹乱し、

第２撹乱用データX2iを用いて前記インデックスに対応したデータを撹乱し、

第４撹乱用データX4iを用いて、前記第２撹乱用データを用いて撹乱されたデータをさらに撹乱することにより生成された第２変形済み表を使い、

前記第３撹乱用データX3iを使って入力データD1を変形し、変形データH1を作成するデータ変形手段と、

前記第１撹乱用データX1iを使って変型データH1を変形し、変形データH2を作成するデータ変形手段と、

変形データH2をインデックスとして、前記第２変形済み表を引いて変形データＨ３を取り出す変形済み表アクセス処理手段と、

処理済変形データＨ３を処理し、処理済変形データＨ４を得る変形済みデータ処理手段と、

第３撹乱用データX3iを用いて処理済み変形データＨ４を変形し、処理済変形データＨ５を得るデータ変形処理手段と、

第１撹乱用データX1iを用いて処理済み変形データＨ５を変形し、処理済変形データＨ６を得るデータ変形処理手段と、

処理済第２撹乱用データX2oを用いて処理済み変形データＨ６を変形し、処理済変形データＨ７を得るデータ変形処理手段と、

処理済第４撹乱用データX4oを用いて処理済み変形データＨ７を変形し、処理済変形データＨ８を得るデータ変形処理手段と、

処理済変形データＨ８をインデックスとして、前記第２変形済み表を引いて変形データＨ９を取り出す変形済み表アクセス処理手段と、

処理済変形データＨ９を処理し、処理済変形データＨ１０を得る変形済みデータ処理手段と、

処理済第２撹乱用データX2oを用いて処理済み変形データＨ１０を逆変形し、変型データＨ１１を得るデータ逆変型処理手段と、

処理済第４撹乱用データX4oを用いて処理済み変形データＨ１１を逆変形し、

入力データD1を用いて表引きを行い、表引き結果をデータ処理処理し、さらにその結果を用いて表引きを行い、表引き結果をデータ処理処理した結果である処理済データD2を得るデータ逆変形処理手段とを有することを特徴とする情報処理装置。

【請求項２４】　第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび第３撹乱用データX3iおよび処理済第４撹乱用データX4oおよび第２変形済み表を生成する手段として、

第１撹乱用データX1iを生成するための第１ハミングウエイト一定乱数生成手

段と、

第２撹乱用データX2iを生成するための第２ハミングウエイト一定乱数生成手段と、

第２撹乱用データX2iを処理し、処理済第２撹乱用データX2oを生成する撹乱用データ処理手段と、

処理済第２撹乱用データX2oのハミングウエイトを計算し、処理済第２撹乱用データX2oのハミングウエイトが不適切な場合は第２ハミングウエイト一定乱数生成手段に対して、

第２撹乱用データX2iの再生成を行わせる、ハミングウエイト検査手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段と、

第３撹乱用データX3iを生成するための第１乱数生成手段と、

第４撹乱用データX4iを生成するための第２乱数生成手段と、

第４撹乱用データX4iを処理し、処理済第４撹乱用データX4oを生成する撹乱用データ処理手段と、

第３撹乱用データX3iと第４撹乱用データX4iと変形済み表とを用いて、変形済み表のインデックスを第３撹乱用データX3iで変形し、変形済み表のデータを第４撹乱用データX4iで変形し、第２変形済み表を生成する表変形処理手段とを有することを特徴とする、請求項２３に記載の情報処理装置。

【請求項２５】　第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび第３撹乱用データX3iおよび処理済第４撹乱用データX4oおよび第２変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第１撹乱用データ格納手段と、

第１撹乱用データ格納手段に格納されたデータをランダムに選択し、第１撹乱用データX1iとするための第１撹乱用データ選択手段と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な値を複数格納している第２撹乱用データ格納手段と、第

２撹乱用データ格納手段に格納されたデータをランダムに選択し、第２撹乱用データX2iとするための第２撹乱用データ選択手段と、

第２撹乱用データX2iを処理し、処理済第２撹乱用データX2oを生成する撹乱用データ処理手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段と、

第３撹乱用データX3iを生成するための第１乱数生成手段と、

第４撹乱用データX4iを生成するための第２乱数生成手段と、

第４撹乱用データX4iを処理し、処理済第４撹乱用データX4oを生成する撹乱用データ処理手段と、

第３撹乱用データX3iと第４撹乱用データX4iと変形済み表とを用いて、変形済み表のインデックスを第３撹乱用データX3iで変形し、変形済み表のデータを第４撹乱用データX4iで変形し、第２変形済み表を生成する表変形処理手段とを有することを特徴とする、請求項２３に記載の情報処理装置。

【請求項２６】　第１撹乱用データX1iおよび処理済第２撹乱用データX2oおよび第３撹乱用データX3iおよび処理済第４撹乱用データX4oおよび第２変形済み表を生成する手段として、

ハミングウエイト一定の値を予め複数格納している第１撹乱用データ格納手段と、

第１撹乱用データ格納手段に格納されたデータをランダムに選択し、第１撹乱用データX1iとするための第１撹乱用データ選択手段と、

ハミングウエイト一定かつ撹乱用データ処理手段で処理した結果のハミングウエイトも一定となる様な、第２撹乱用データと処理済第２撹乱用データの組を複数組格納している第２撹乱用データおよび処理済第２撹乱用データ格納手段と、第２撹乱用データおよび処理済第２撹乱用データ格納手段に格納された、第２撹乱用データと処理済第２撹乱用データの組をランダムに選択し、第２撹乱用データX2iおよび処理済第２撹乱用データX2oとするための第２撹乱用データおよび処理済第２撹乱用データ選択手段と、

第１撹乱用データX1iと第２撹乱用データX2iと表を用いて、表のインデックスを第１撹乱用データX1iで変形し、表のデータを第２撹乱用データX2iで変形し、変形済み表を生成する表変形手段と、

第３撹乱用データX3iを生成するための第１乱数生成手段と、

第４撹乱用データX4iを生成するための第２乱数生成手段と、

第４撹乱用データX4iを処理し、処理済第４撹乱用データX4oを生成する撹乱用データ処理手段と、

第３撹乱用データX3iと第４撹乱用データX4iと変形済み表とを用いて、変形済み表のインデックスを第３撹乱用データX3iで変形し、変形済み表のデータを第４撹乱用データX4iで変形し、第２変形済み表を生成する表変形処理手段とを有することを特徴とする、請求項２３に記載の情報処理装置。

【請求項２７】　プログラムを格納するプログラム格納部、データを保存するデータ格納部を有する記憶装置と

プログラムにしたがい、所定の処理を実行し、データ処理を行なう中央演算装置を有し、

プログラムは、中央演算装置の指示を与える処理命令から構成される一つ以上のデータ処理手段を有し、

一つのデータ処理手段が、メッセージと秘密鍵を入力とし、Data Encryption Standard(DES)暗号処理を行い、暗号化処理を行ない、前記暗号化処理の処理結果を生成し、且つ、

平文を撹乱する平文撹乱用データPXを用いて、メッセージを変形する手段と、

秘密鍵を撹乱する秘密鍵撹乱用データKXを用いて、秘密鍵データＫを変形する手段と、

ＤＥＳ暗号処理で用いられるＳＢＯＸ表のインデックスを撹乱するためのＳＢＯＸアドレス撹乱用データSinX1を用いてSBOX表のインデックスを変形しＳＢＯＸ表を並べ替え、かつＳＢＯＸ表の出力を撹乱するためのＳＢＯＸデータ撹乱用データSoutXとを用いて並べ替えられたＳＢＯＸ表の出力を撹乱し、変形済みＳＢＯＸ表を作成するためのＳＢＯＸ表変形手段と、

ＳＢＯＸの表引き処理の直前のＸＯＲ結果が、データがＳｉｎＸ１および平文

撹乱用データＰＸもしくはＰＸを変形した値で変形された値となるように調整するための逆変形処理もしくは変形処理を、ＸＯＲの２つの入力データのいずれかもしくは両方に有し、

ＳＢＯＸの表引き処理の前までにデータがＳｉｎＸ１で変形された値に調整するための、逆変形処理をＳＢＯＸの直前に有し、

ＤＥＳの最終ラウンド終了後のＩＰ置換処理の直前もしくは直後に、平文撹乱用データＰＸもしくはＰＸを変形された値を逆変形するための逆変形処理手段を有することを特徴とする、情報処理装置。

【請求項２８】　前記ＳＢＯＸ表アドレス撹乱用データＳｉｎＸ１のハミングウエイトが一定かつ、ＳＢＯＸ表データ撹乱用データＳｏｕｔＸのハミングウエイトと、転置済みＳＢＯＸデータ撹乱用データＸＳｏｕｔＸのハミングウエイトが一定であるような、ＳｏｕｔＸを用いていることを特徴とする、請求項２７に記載の情報処理装置。

【請求項２９】　前記ＳＢＯＸ表アドレス撹乱に２つ以上の撹乱用データ用いて、アドレスの変形を複数回おこない、

ＳＢＯＸ表のデータ撹乱に２つ以上の撹乱用データを用いて、変形を複数回以上行なうＳＢＯＸ表変形手段を有することを特徴とする、請求項２７に記載の情報処理装置。

【請求項３０】　ＳＢＯＸ表アドレス撹乱用データのうち、１つ以上の撹乱用データのハミングウエイトのが一定かつ、ＳＢＯＸ表データ撹乱用データのうち、１つ以上の撹乱用データのハミングウエイトが一定であるような、撹乱用データを用いることを特徴とする、請求項２９に記載の情報処理装置。

【請求項３１】　ハミングウエイトが一定となる撹乱用データが、中央演算装置で一度に処理できるビット長に分割した際のそれぞれの部分ビットにおけるハミングウエイトも一定であるような、ＳＢＯＸアドレス撹乱用データＳｉｎＸ１、ＳＢＯＸデータ撹乱用データＳｏｕｔＸ１を用いることを特徴とする、請求項３０に記載の情報処理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、機密性の高いＩＣカードなどの耐タンパ装置に関するものである。

【０００２】

【従来の技術】

ＩＣカードは、勝手に書き換えることが許されないような個人情報の保持や、秘密情報である暗号鍵を用いたデータの暗号化や、或いは暗号文の復号化を行う装置である。ＩＣカード自体は電源を持っておらず、ＩＣカード用のリーダライタに差し込まれると、電源の供給を受け、動作可能となる。ＩＣカードは動作可能になると、リーダライタから送信されるコマンドを受信し、そのコマンドに従って、データの転送等の処理を行う。

【０００３】

ＩＣカードの基本概念は、図１に示すように、カード１０１の上に、ＩＣカード用チップ１０２を搭載したものである。図に示すように、一般にＩＣカードは、所定位置に、供給電圧端子Ｖｃｃ、グランド端子ＧＮＤ、リセット端子ＲＳＴ、入出力端子Ｉ／Ｏ、クロック端子ＣＬＫ及び、を有する。この端子の位置は、ＩＳ０７８１６の規格に定められている。これらの諸端子を通して、リーダーライタから電源の供給やリーダライタとのデータの通信を行う。こうしたＩＣカードを用いた通信に関しては、例えばW．Ｒａｎｋｌ Ａｎｄ Ｅｆｆｉｎｇ：ＳＭＡＲＴＣＡＲＤ ＨＡＮＤＢＯＯＫ、Ｊｏｈｎ Ｗｉｌｅｙ ＆ Ｓｏｎｓｓ、１９９７、ＰＰ.４１などに見られる。

【０００４】

ＩＣカードに搭載される半導体チップの構成は、基本的には通常のマイクロコンピュータと同じ構成である。図２はＩＣカードに搭載される半導体チップの基本的構成を示すブロック図である。図２に見られるように、カード部材用の半導体チップは、中央処理装置（ＣＰＵ）２０１、記憶装置２０４、入出力（Ｉ／Ｏ）ポート２０７、コ・プロセッサ２０２を有する。システムによってはコ・プロセッサはない場合もある。ＣＰＵ２０１は、論理演算や算術演算などを行う装置であり、記憶装置２０４は、プログラムやデータを格納する装置である。入出力ポートは、リーダライタと通信を行う装置である。コ・プロセッサは、暗号処理

そのもの、または、暗号処理に必要な演算を高速に行う装置である。これには、例えば、ＲＳＡ暗号の剰余演算を行う為の特別な演算装置や、ＤＥＳ暗号のラウンド処理を行う暗号装置などがある。ＩＣカード用プロセッサの中には、コ・プロセッサを持たないものも多くある。データバス２０３は、各装置を接続するバスである。

【０００５】

記憶装置２０４は、ＲＯＭ（Ｒｅａｄ　Ｏｎｌｙ　Ｍｅｍｏｒｙ）やＲＡＭ（Ｒａｎｄｕｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）、ＥＥＰＲＯＭ（Ｅｌｅｃｔｒｉｃ　Ｅｒａｓａｂｌｅ　Ｐｒｏｇｒａｍａｂｌｅ　Ｒｅａｄ　Ｏｎｌｙ　Ｍｅｍｏｒｙ）などを有する。ＲＯＭは、記憶情報を変更できないメモリであり、主にプログラムを格納するメモリである。ＲＡＭは自由に書き換えができるメモリであるが、電源の供給が中断されると、記憶している内容は消滅する。ＩＣカードがリーダライタから抜かれると電源の供給が中断されるため、ＲＡＭの内容は、保持されなくなる。ＥＥＰＲＯＭは、電源の供給が中断されてもその内容を保持することができるメモリである。このＥＥＰＲＯＭは記憶情報を書き換える必要があり、ＩＣカードがリーダライタから抜かれても、保持が可能なデータを格納するために使われる。例えば、プリペイドカードでのプリペイドの度数などは、使用するたびに書き換えられ、かつリーダライタか抜かれてもデータを保持する必要があるため、ＥＥＰＲＯＭに保持される。

【０００６】

ＩＣカードは、プログラムや重要な情報がＩＣカード用チップの中に密閉されているため、重要な情報を格納したり、カードの中で暗号処理を行うために用いられる。従来、ＩＣカードでの暗号を解読する難しさは、暗号アルゴリズムの解読の困難さと同じと考えられていた。しかし、ＩＣカードが暗号処理を行っている時の消費電流を観測し、解析することにより、暗号アルゴリズムの解読より容易に暗号処理の内容や暗号鍵が推定される可能性が示唆されている。消費電流は、リーダライタから供給されている電流を測定することにより観測することができ、この攻撃法の詳細は、例えば、Ｊｏｈｎ　Ｗｉｌｅｙ　＆　Ｓｏｎｓ社、Ｗ．Ｒａｎｋｌｅ　＆　Ｗ．Ｆｆｉｎｇ著「ＳＭＡＲＴ　ＣＡＲＤ　ＨＡ

ＮＤＢＯＯＫ」の８．５．１．１ Ｐａｓｓｉｖｅ Ｐｒｏｔｅｃｔｉｖｅ Ｍｅｃｈａｎｉｓｍｓ（２６３ページ）にこのような危険性が記載されている。それは次のような理由による。　ＩＣカード用チップを構成しているＣＭＯＳは、出力状態が１から０あるいは０から１に変わった時に電流を消費する。特に、データバス２０３においては、バスドライバーの電流や、配線及び、配線に接続されているトランジスタの静電容量のため、バスの値が１から０あるいは０から１に変わると、大きな電流が流れる。そのため、消費電流を観測すれば、ＩＣカード用チップの中で、何が動作しているか分かる可能性を示唆している。

【０００７】

図３は、ＩＣカード用チップの１サイクルでの消費電流の波形を示したものである。処理しているデータに依存して、電流波形が３０１や３０２のように異なる。このような差は、バス２０３を流れるデータや中央演算装置２０１で処理しているデータに依存して生じる。

【０００８】

16ビットのプリチャージバスにデータを転送する場合を考える。プリチャージバスは、データ転送の前に全てのビットを”０”にそろえるバスである。このバスに、値は違うが、”１”のビットの数が同じデータ、たとえば、”１”のビットの数が２である16進数の”８８”と”１１”を転送した場合、電流波形はほぼ同じ波形になる。この理由は、”０”から”１”へ変化したビットの数が同じであるため、同じように電流を消費し、同じ電流波形になったからである。もし、”１”のビットの数が１つ異なるデータ、たとえば”１”のビットの数が３である”８９”や”１９”を転送した場合、”１”のビットの数が２のデータとは消費電流波形が異なる。これは、３ビット分のバスの値が”０”から”１”に変化したため、その分の電流が消費される。そのため、先の２ビットが変化したデータに比べて、消費電流が１ビット分大きくなる。一般に、”１”のビットの数が多いほど、電流波形は高くなると言う規則性がある。この規則性から、転送されるデータを推定することができる。

【０００９】

具体的な命令でどのように差が出るかを、次のような左シフト命令を例にして

説明する。

ｌｏｇｉｃａｌ＿ｓｈｉｆｔｌ　Ｒ１　　（式１）

この命令は、レジスタ１の内容を左シフトし、最上位ビットの値をコンディションコードレジスタのキャリーフラグに入れる命令である。レジスタＲ１の最上位ビットが内部バスを経由してコンディションコードレジスタに転送されるため、電流波形の大きさを比較すれば、最上位ビットの”０”と”１”が識別できる可能性がある。もしＲ１に重要なデータが入っていれば、そのデータの1ビットが”０”であるか”１”であるかが分かる可能性がある。特にＤＥＳのような暗号処理では、暗号鍵をシフトする操作が頻発する。このシフト操作の時に暗号鍵のデータを推定できる電流波形が生じ、暗号鍵を推定される危険性がある。

【００１０】

これは、コ・プロセッサ２０２の演算でも同じである。演算内容が暗号鍵に依存した偏りがあると、その偏りが消費電流から求められ、暗号鍵が推定される可能性がある。

【００１１】

特願平10-354156号の「情報処理装置、端タンパ処理装置」では、この課題を解決する方法として、撹乱用データで処理データを変形し、変形されたデータを処理し、処理結果を撹乱用データを用いて逆変換することにより、処理中の消費電流とデータの関連性を減らすことで解決しようとしている。

【００１２】

次の命令列を例として、問題点を説明する。

ｌｏｇｉｃａｌ＿ｒｏｔａｔｅｌ　Ｒ１　　（式２）

ｘｏｒ　Ｒ１　Ｒ２　　（式３）

（式２）では、レジスタＲ１の値を左に論理ローテートし、レジスタＲ１に格納する。（式３）では、レジスタＲ２とＲ１の排他的論理和を取り、Ｒ２に格納する。（式２）と（式３）では、処理するデータをそのまま扱っているので、データの内容によって消費電流波形の大きさが変わり、電流波形を観測することによって、データが推定できる。

特開平10-354156号では、問題を解決するために、勝手に選んだ乱数Ｘ１、Ｘ２

を撹乱用データとして用い、以下に示すように（式４）、（式５）でＲ１、Ｒ２に格納されたデータを変形し、（式６）、（式７）で変形データを処理し、Ｒ２に格納する。（式８）、（式９）で逆変形するための準備計算を行い、（式１０）でＲ２に格納された処理済変形データを逆変形し、（式２）（式３）で得られるものと同じ値がＲ２に格納される。

```
ｘｏｒ    Ｘ１  Ｒ１                    （式４）
ｘｏｒ    Ｘ２  Ｒ２                    （式５）

ｌｏｇｉｃａｌ＿ｒｏｔｅｔｅｌ  Ｒ１    （式６）
ｘｏｒ    Ｒ１  Ｒ２                    （式７）

ｌｏｇｉｃａｌ＿ｒｏｔｅｔｅｌ  Ｘ１    （式８）
ｘｏｒ  Ｘ１  Ｘ２                      （式９）

ｘｏｒ  Ｘ２  Ｒ２                      （式１０）
```

ここで問題となるのは、特開平10-354156号では、データのハミングウエイトが直接観測されないように、撹乱用データを使っていた。しかし、撹乱用データはある確率でハミングウエイトが０や８（特別な値）になる。もし、撹乱用データがそのような特別のハミングウエイトを持つと、実際の処理データのハミングウエイトが直接観測できることになる。本発明では、撹乱用データのハミングウエイトを０や８などの特別な値に成ることを防ぐものである。

【００１３】

具体的には、（式４）、（式５）を計算する際に、Ｘ１、Ｘ２の値に依存した消費電流の差が観測可能であり、Ｘ１、Ｘ２のハミングウエイトが推定できる。たとえば、Ｘ１、Ｘ２のハミングウエイトと消費電流が比例するプロセッサの場合、ハミングウエイトが０となる場合を検出できる。同様に、ｘｏｒ演算でのビットがする反転数と消費電流が比例する場合においても、反転するビット数は、Ｘ１、Ｘ２のハミングウエイトと等しい。ハミングウエイトが０となる値は０以

外にはありえないため、消費電流を観測することで、撹乱用データが０となる場合の測定データのみを識別することが出来る。上記の撹乱方式では、（式６）（式７）の計算に際して、（式２）（式３）と同じ電流波形が観測されることなり、アタックが可能となる。

【００１４】

【発明が解決しようとしている課題】

本願発明は、高いセキュリティを持つカード部材などの耐タンパー情報処理装置を提供するものである。

【００１５】

本願発明の技術的な課題は、カード部材、例えばＩＣカード用チップでのデータ処理と消費電流との関連性を減らすことである。消費電流とチップの処理との関連性が減れば、観測した消費電流の波形からＩＣカード用チップ内での処理や暗号鍵を推測することが困難になる。即ち、本願発明は、カード部材等に高いセキュリティを持たせんとするものである。

【００１６】

【課題を解決するための手段】

本願発明の着眼点は、ＩＣカード用チップで消費される電流値と、処理されているデータの関連性を減らすための方法として、処理するデータを撹乱用データで変形し、データの処理を変形したデータで処理し、処理後に撹乱用データを用いて逆変換し、正しい処理結果を求めるものである。さらに、撹乱用データと消費電流の関連性を減らすために、撹乱用データとして、データを変形する際に用いる撹乱用データのバイナリ表現した際のハミングウエイトが常に一定値となりかつ各ビットが”０”と”１”のどちらの値をとるのかの確率が0.5となる。且つ、逆変換する際に用いる撹乱用データのバイナリ表現もまたハミングウエイトが一定となり、各ビットが”０”と”１”のどちらの値をとるのかの確率が0.5であるような撹乱用データを生成することにより、撹乱用データを用いた処理の消費電流と撹乱データの関連性を減らし、撹乱用データを消費電流から推定した後に、変形したデータを消費電流から推定し、推定された撹乱用データと変形したデータから、元の処理データを推定するアタックを困難にする。尚、ここで、

ハミングウエイトとは、二進数における“１”の数を意味する。

【００１７】

さらに、撹乱用データを撹乱用データの生成方法として、あらかじめ撹乱データとして用いることの出来る値を複数生成し、格納しておくことにより、撹乱用データを生成する際に撹乱用データを生成する際の消費電流と撹乱用データの関連性を減らし、撹乱用データを推定することを困難にする。

【００１８】

【発明の実施の形態】

以下、本発明の実施例に付いて図面を参照しながら説明する。

【００１９】

図１はICカードの概観を示したものである。ICカード１０１は、ISO７８１６の規格により、大きさや、ICカードのチップ１０２の位置や接点の数および割り当てなどが規定されている。

【００２０】

図２はICカードチップ１０２の内部構成である。構成に付いては、従来技術の説明で既に述べた通りである。本発明は、プログラム２０５で処理するデータに撹乱を加えることにより、処理中に生じるICカード用チップのハードウエアが消費する電流波形から、本物のデータの推定を困難にさせるものである。

【００２１】

従来の技術で説明したように、データをそのまま処理した場合、消費電流を測定することにより、データを推定することが可能である。また、処理されるデータを撹乱用のデータを用いて変形し、変形したデータに対して処理を行い、選られた処理結果を撹乱用に用いたデータもしくは撹乱用に用いたデータを処理した値を用いて逆変形を行い、本物の処理後データと等しい値を得ることにより、処理中に消費される電流値とデータとの間の相関を減らし、消費電流の測定からデータを推定することを困難にする従来技術では、撹乱に用いるデータについて制限を設けていないため、撹乱用データが処理させる際の消費電流値により、撹乱用データを推定し、測定データを分類することで、なおアタックが可能である。

【００２２】

例えば、XORを撹乱用の関数として用いた場合、撹乱用データの全ビットが１の場合や全ビットが０の場合などは、消費電流により識別可能であり、また識別率が１００％でない場合でも、多くの測定サンプルの平均値を計算することにより、識別誤りの影響をデータの推定には影響が無くなるようにすることが出来る。

【００２３】

尚、前記処理の例は、例えば、ローテート、シフト、ビット位置置換、拡大ビット位置置換などの処理をあげる事が出来る。

【００２４】

そこで、撹乱用データのハミングウエイトが撹乱用データのビット長の半分に等しくなり、かつ撹乱用データのそれぞれのビット位置での０と１の出現確率が0.5となるように撹乱用データを生成することで、撹乱用データが処理される際の消費電流より、撹乱用データが容易に特定できなくなる。

【００２５】

入力データをＤ１、処理を関数ｆ、出力データをＤ２とするとき、

Ｄ２＝ｆ（Ｄ１）　　（式１１）

となる。処理ｆの電流波形を測定することで、Ｄ１が推定可能とする。ここで、撹乱用データＸ１ｉを用いて、Ｄ１を撹乱する変形関数ｈと、ｈの逆変換を行うｇという変換関数があり、次の（式１２）もしくは（式１３）の関係を満たす場合、（式１１）の代わりに（式１２）もしくは（式１３）を計算することで、（式１１）相当の処理を行う。

【００２６】

（式１２）もしくは（式１３）のいずれが使用できるのかは、関数ｆ、関数ｈの性質に依存する。（式１２）を満たす、関数ｆ、関数ｈの例としては、関数ｆがローテート、シフト、ビット位置置換などの処理（式１４）であり、関数ｈがｘｏｒ（式１５）の場合が例としてあげられる。その場合は、関数ｇもまた、ｘｏｒ（式１６）となる。

【００２７】

一方、（式１３）を満たす、関数ｆ、関数ｈの例としては、関数ｆが加減算で、関数ｈも加減算の場合、関数ｆが乗除算で、関数ｈも乗除算の場合が例として上げられる。

（式１２）や（式１３）においても、処理ｆの電流波形を測定することで、ｈ（Ｄ１、Ｘ１ｉ）の値は推定可能であるが、Ｘ１ｉが推定できないと、Ｄ１を復元することはできない。

ｆ（Ｄ１）＝ｇ（ｆ（ｈ（Ｄ１、Ｘ１ｉ））、ｆ（Ｘ１ｉ））（式１２）

ｆ（Ｄ１）＝ｇ（ｆ（ｈ（Ｄ１、Ｘ１ｉ））、Ｘ１ｉ）　（式１３）

ｈ（ｘ、ｙ）＝ｘ　ｘｏｒ　ｙ　（式１４）

ｆ（ｘ）＝ｒｏｔａｔｅ＿ｒｉｇｈｔ（ｘ）　（式１５）

ｇ（ｘ、ｙ）＝ｘ　ｘｏｒ　ｙ　（式１６）

ところが、Ｘ１ｉが値が特定の値Ｃになった場合に、外部からの観測で識別可能で、変形関数ｈが既知である場合、ｈ（Ｄ１、Ｃ）の値から、ｈの逆関数を取ることで、Ｄ１を復元出来る。外部からの電流測定により識別が容易なＸ１ｉとしては、すべてのビットが０である場合や、すべてのビットが１の場合などが有る。というのは、ハミングウエイトが０となるデータは０以外には存在せず、同様にすべてのビットが１となる場合のハミングウエイトと等しいハミングウエイトを与える値は、すべてのビットが１となるもの以外にはないからである。関数ｈとして、ｘｏｒを用いた場合、Ｘ１ｉが０となった場合、ｈ（Ｄ１、０）が即Ｄ１の値と等しくなる。値のバラエティーが最も多いハミングウエイトは、データのビット長の半分のハミングウエイトである。

【００２８】

図４は、１個の撹乱用データを用いた実施例である。本実施例の特徴は、撹乱用データのハミングウエイトを与えることによって、撹乱用データの各ビットが

オール０になったり、オール１になったりすることにより、データが推定されることを防ぐものである。データ変形処理手段（４０２）で入力データＤ１（４０１）を撹乱用データX1i（４０３）を用いて変形し、変形データＨ２（４０４）を生成し、変形データ処理手段（４０５）にて、変形データＨ１（４０４）を処理し、変形データＨ２（４０６）を得る。データ逆変換処理手段（４０７）で、変形データＨ２（４０７）を変形済み撹乱用データＸ１ｏ（４０８）を用いて逆変換し、処理済データＤ２（４０９）を得る。ここで、撹乱用データＸ１ｉ（４０３）と変形済み撹乱用データＸ１ｏ（４０８）は、ともにハミングウエイトが一定の値である。

【００２９】

この実施例は、請求項１に対応する実施例である。撹乱用データＸ１ｉ（４０３）と、変形済み撹乱用データＸ１ｏ（４０８）を作成する方法には幾つかの方法が有る。

【００３０】

図５は、ハミングウエイト一定の撹乱用データＸ１ｉ（５０２）およびハミングウエイト一定の処理済撹乱データＸ１ｏ（５０４）を生成する一実施例である。ハミングウエイト一定乱数生成手順（５０１）により、ハミングウエイト一定の乱数を生成し、第１撹乱用データＸ１ｉ（５０２）とし、撹乱用データ処理手段（５０３）で第１撹乱用データＸ１ｉ（５０２）を処理し、処理済撹乱用データＸ１ｏ（５０４）を得る。ハミングウエイト検査手段（５０５）により、処理済撹乱用データＸ１ｏ（５０４）のハミングウエイトが検査され、所定のハミングウエイトと異なる場合は、ハミングウエイト一定の乱数生成手段（５０１）に対して再生成制御信号を送り、撹乱用データＸ１ｉ（５０２）の生成からやりなおす。これは、請求項２の実施例である。前記ハミングウエイト検査は、具体的には多くはＣＰＵで実行される。ハミングウエイト一定の乱数生成手段（５０１）はＣＰＵあるいはその生成手段（Ｇｅｎｅｒａｒｔｏｒ）がその役割を果たす。

【００３１】

ハミングウエイト一定乱数生成の方法はいくつか存在する。図６は、ハミング

ウエイト一定の乱数の生成方法の１実施例で、請求項６の実施例でもある。ここでは、発生したい乱数のビット数を２ｎビットとする。まずｎビット乱数発生手段（６０１）によりｎビット乱数（６０２）を生成する。このｎビット乱数発生手段（６０１）は、疑似乱数でもよいし、物理現象を測定して選られる真の乱数であってもよい。つぎにビット反転処理手段（６０３）を用いて、発生したｎビット乱数（６０２）を反転し、反転ｎビット乱数（６０４）を生成する。つぎに、データ統合手段（６０５）にて、ｎビット乱数（６０２）と反転ｎビット乱数（６０４）を結合して、ハミングウエイト一定２ｎビット乱数（６０６）を生成する。ｎビット乱数（６０２）の各ビットのうち、値が１であるビット数をｎ１個、０であるビット数をｎ２個とすると、

ｎ１＋ｎ２＝ｎ　　（式１７）

であり、反転ｎビット乱数（６０４）はｎビット乱数（６０２）の反転値であるので、値が１であるビットの数はｎ２個、０であるビット数がｎ１個となる。従って、ｎビット乱数（６０２）と反転ｎビット乱数（６０４）を結合して作成した乱数のハミングウエイトは、ｎ１＋ｎ２となり、（式１７）に示されるように、ここで生成される乱数のハミングウエイトは、常にｎビットとなる。

【００３２】

図７はハミングウエイト一定乱数生成の別の実施例で、請求項７の実施例である。目標のハミングウエイトＨを受け取り（７０２）、乱数Ｒを発生させる（７０３）。乱数ＲのハミングウエイトＲＨを計算し（７０４）、発生させた乱数ＲのハミングウエイトＲＨが目標のハミングウエイトＨと等しいか比較し、等しくない場合は、乱数Ｒ発生（７０３）からやり直す。乱数ＲのハミングウエイトＨと、目標のハミングウエイトＨが等しい場合は、乱数Ｒを結果として返し（７０６）、処理を終了する（７０７）。

【００３３】

図１０は、ハミングウエイト一定乱数生成の別の実施例で、請求項８の実施例である。予め、ｍビット長のデータで、ハミングウエイトが一定であるデータを

収めたテーブルを用意しておく。発生できるハミングウエイト一定の乱数は、ｍの倍数ビット長にかぎられる。まず、発生したい乱数のビット長をｎに設定し（１００２）、Ｌにｎをｍで割った値を代入する（１００４）。処理の基本的な流れは、ｍビット長のハミングウエイト一定の乱数をＬ個生成し結合することで、ｎビット長のハミングウエイト一定の乱数を生成する。つぎに、ハミングウエイト一定の乱数の生成結果を収めるＤを０に初期化する（１００４）。乱数Ｒを生成し、乱数Ｒをインデックスとして、あらかじめハミングウエイトが一定であるデータを収めたテーブルから値を１つ取り出し、ｄに代入する（１００６）。Ｄをｍビット分左シフトし、ｄを加える（１００７）。乱数Ｒ発生（１００５）からＤをｍビット分左シフトし、ｄを加える処理（１００７）までを、Ｌ回繰り返し（１００８）、Ｄを結果として返す（１００９）。

【００３４】

図８は、決められたビット長（MaxBit）のデータで、ハミングウエイトがHammingビット（一定のハミングウエイト）となるデータをすべてリストアップし、datという配列に入れるための処理手順の実施例である。dat配列の大きさは、（MaxBitの階乗）／｛（Hammingの階乗）＾２｝になる。尚、ここで、ｄａｔ配列とは、ハミングウエイトが一定のデータを格納するための配列である。たとえば、８ビット長のデータで、ハミングウエイトが４ビットとなるようなデータは、７０個になる。基本的な考え方は、Hamming個の１となるビット位置を格納する配列を用意し、その配列におさめられたビット位置が重ならないように１となるビット位置を変えて行くことで、すべての組み合わせを見つけ出すというものである。まず、ハミングウエイトをHammingに代入する（８０２）。つぎに生成したいデータのビット長をMaxBitに代入する（８０３）。次にHamming個の要素を持つ、１となるビットの位置を格納するpos配列を０からHamming-1までの値で初期化する（８０４）。結果を格納するためのdat配列へのインデックスnumを０に初期化する（８０５）。尚、ここで、ｎｕｍは配列ｄａｔのインデックス値で、処理（８０６）で生成される値を格納するインデックスの位置を示す。まず、pos配列に収められたビット位置から、データを計算し、dat配列のインデックスがnumの位置に収める（８０６）。インデックスnumに１を加える（８０７）。更新

するビット位置を決めるインデックスｂを０に初期化する（８０８）。ｂが一番上位に位置するビットにたどり着いているかチェックし（８０９）、たどり着いていなければ、処理８０１に、たどり着いていれば、処理８１２に分岐する。処理８０１では、１ビット上位がすでに１になっているかチェックし、すでに１があれば、処理８１２に飛び、１がなければ、処理８１１へ移行する。処理８１１では、ビットの位置を１ビット分上位方向に進め、処理８０６に移る。処理８１２では、現在注目しているビットが、１となっているビットのうちの最上位のビットかどうかをチェックし、最上位であった場合には、処理８１３に進み、それ以外の場合は処理８１４に進む。処理８１３では、１ビット上位にビット移動できるかチェックし、移動できる場合は、８１１に飛び、移動できない場合は、処理８１４に飛ぶ。処理８１４では、現在注目しているビットが値が１となっているうちの最下位のビットかどうかをチェックし、最下位のビットであれば処理８１５に飛び、最下位ではない場合は処理８１６に飛ぶ。処理８１６では、下位側のビットのうち、１になっているビットで一番近くにあるビットから数えて、１ビット分上位ビットの位置に移動させる。処理８１５では、最下位にビットを移動させ、処理８１７に移る。処理８１７では、注目するビットを１つ分、上位方向に位置するビットに移す。処理８１８では、全部の組み合わせについて処理が終了したかをチェックし、まだ終わっていなければ、処理８０６に飛び、すべての組み合わせに付いて処理が終わっていれば、終了する（８１９）。結果は、dat配列に格納され、dat配列に格納されたデータの個数は、numに設定される。

【００３５】

図１１は、撹乱用データＸ１ｉ（１１０３）、変形済み撹乱用データＸ１ｏ（１１０５）を生成する一実施例で、請求項３の実施例である。予め撹乱用データＸ１ｉとして使用可能なデータを格納した撹乱用データ格納手段（１１０２）より、撹乱用データ選択手段（１１０１）がデータを選択し、撹乱用データＸ１ｉ（１１０３）とする。選択された撹乱用データＸ１ｉ（１１０３）を撹乱用データ処理手段（１１０４）により処理し、変形済み撹乱用データＸ１ｏ（１１０５）を生成する。撹乱用データ格納手段は通例、例えばＲＡＭあるいはレジスタなどが、又撹乱用データ処理手段は通例、例えばＣＰＵあるいはＡＬＵなどが用い

られる。図９は、撹乱用データ格納手段（１１０２）に格納するための、撹乱用データを予め選択する方法の一実施例である。

【００３６】

図９の実施例は、図８の実施例と同様の手順で、ハミングウエイト一定のデータをリストアップしている。図８の実施例との違いは、リストアップされたデータをそのまま使用するのではなく、さらに撹乱用データ処理手段で処理し、処理結果のハミングウエイトを計算してhxdatに代入し（９０７）、ｈｘｄａｔの値もハミングウエイト一定となることをチェックし（９０８）、一定の場合だけ、dat配列に格納するようにしている点である。その他の処理に付いては、図８と同一である。

【００３７】

図１２は、撹乱用データＸ１ｉ（１２０３）、変形済み撹乱用データＸ１ｏ（１２０４）を生成する一実施例で、請求項４の実施例である。予め、図９の実施例や図５の実施例などの方法で、撹乱用データおよび処理済撹乱用データ格納手段（１２０１）に撹乱用データおよび処理済撹乱用データの組を複数組格納し、撹乱用データおよび処理済撹乱用データ選択手段（１２０２）により、撹乱用データおよび処理済撹乱用データ格納手段（１２０１）から、撹乱用データＸ１ｉ（１２０３）及び処理済撹乱用データＸ１ｏ（１２０４）を取り出す。図５５に図１２に対応するテーブルの例を示す。これはデータ処理が、左ローテートの場合で、前記撹乱用データおよび処理済撹乱用データ格納手段（１２０１）に格納されたデータＸ１ｉ、Ｘ１ｏを例示する。

【００３８】

また、撹乱用データおよび処理済撹乱用データ格納手段（１２０１）に格納するデータの個数を偶数にし、格納するデータを適切に選択することで、請求項５の実施例となる。請求項５の実施例としては、最低限２組のデータがあればよい。

【００３９】

図１３は、データ処理がテーブルルックアップで定義された処理を、２個の撹乱用データを用いて撹乱する実施例で、請求項９の一実施例である。入力データ

Ｄ１（１３０１）を用いて、表を引き、処理済データＤ２（１３１０）得る。表をＴａｂｌｅとすると、Ｄ１とＤ２の関係は、（式１９）のようになる。

Ｄ２＝Ｔａｂｌｅ[Ｄ１]　　（式１９）

この表引き処理を行う際の電流を測定すると、Ｄ１やＤ２の値が推定できる。そこで、第１撹乱用データＸ１ｉと第２撹乱用データＸ２ｉと、表のインデックスと出力結果を変形する関数ｆと関数ｇを考え、変形済みの表ＸＴａｂｌｅを（式２０）のように定義する。

ＸＴａｂｌｅ[ｆ（ｉ、Ｘ１ｉ）]
＝ｇ（Ｔａｂｌｅ[ｉ]、Ｘ２ｉ）　　（式２０）

また、ｇの逆関数をｈと定義する。

Ｄ＝ｈ（ｇ（Ｄ、Ｘ）、Ｘ）　　（式２２）

すると、表引き処理は、

Ｈ１＝ｆ（Ｄ１、Ｘ１ｉ）　　（式２３）
Ｈ２＝ＸＴａｂｌｅ[Ｈ１]　　（式２４）
Ｄ２＝ｈ（Ｈ２、Ｘ２ｉ）　　（式２５）

となる。関数ｆ（ｘ、ｙ）は、ｘの値が異なる場合に、必ず異なる値になる必要がある。一方、関数ｇ、関数ｈは、（式２６）の関係を満たす必要がある。

ａ＝ｈ（ｇ（ａ、Ｘ）、Ｘ）　　（式２６）

ここで、（式２４）の処理の消費電流によりＨ１もしくはＨ２が推定されても、

それぞれＸ１ｉ、Ｘ２ｉにより撹乱されているため、（式２４）の測定結果だけでは、Ｄ１、Ｄ２の値を推定することは出来ない。図１３では、（式２３）にあたる処理が、第１撹乱用データＸ１ｉ（１３０３）を用いて、入力データＤ１（１３０１）をデータ変形処理手段（１３０２）で処理し、変形データＨ１（１３０４）を得る処理に相当する。また、（式２４）に当たる処理は、変形データＨ１（１３０４）をインデックスとして、変形済み表（１３０６）を表引きする変形済み表アクセス手段（１３０５）を用いて、変形データＨ２（１３０７）を得る処理に相当する。（式２５）に相当する処理は、変形データＨ２（１３０７）を第２撹乱用データＸ２ｉ（１３０９）を用いてデータ逆変形処理手段（１３０８）を用いて逆変形し、処理済データＤ２（１３０１）を得る処理に相当する。

【００４０】

図１４は、図１３の実施例で用いる、第１撹乱用データＸ１ｉ（１４０３）および第２撹乱用データＸ２ｉ（１４０４）および変形済み表（１４０７）を作成するための一実施例で、請求項１０の一実施例である。第１ハミングウエイト一定乱数生成手段（１４０１）により、第１撹乱用データＸ１ｉ（１４０３）を生成し、第２ハミングウエイト一定乱数生成手段（１４０２）により、第２撹乱用データＸ２ｉ（１４０４）を生成し、表格納手段（１４０５）に格納された、（式１９）を満たす表と、第１撹乱用データＸ１ｉ（１４０３）と、第２撹乱用データＸ２ｉ（１４０４）から、表変形手段（１４０６）により（式２０）を満たす変換が行なわれ、変形済み表（１４０７）を生成する。第１ハミングウエイト一定乱数生成手段（１４０１）および第２撹乱用データＸ２ｉ（１４０４）には、請求項６、請求項７、請求項８のいずれかのハミングウエイト一定乱数生成手段を用いることが出来る。

【００４１】

図１５は、図１３の実施例で用いる、第１撹乱用データＸ１ｉ（１５０５）および第２撹乱用データＸ２ｉ（１５０６）および変形済み表（１５０９）を作成するための一実施例で、請求項１１の一実施例である。予めハミングウエイト一定のデータが複数個格納された、第１撹乱用データ格納手段（１５０１）と、第１撹乱用データ格納手段（１５０１）から値を１つ選択して取り出し、第１撹乱

用データＸ１ｉ（１５０５）とするための第１撹乱用データ選択手段（１５０３）と、第２撹乱用データ格納手段（１５０２）から値を１つ選択して取り出し、第２撹乱用データＸ２ｉ（１５０６）とするための第１撹乱用データ選択手段（１５０４）と、表格納手段（１５０７）に格納された、（式１９）を満たす表と、第１撹乱用データＸ１ｉ（１５０５）と、第２撹乱用データＸ２ｉ（１５０６）から、表変形手段（１５０８）により（式２０）を満たす変換が行なわれ、変形済み表（１４０７）が生成される。

【００４２】

図５６に第１撹乱用データ格納手段（１５０１）に格納された第１撹乱用データ、及び第２撹乱用データ格納手段（１５０２）に格納された第２撹乱用データの例を示す。図５７に、表格納手段（１５０７）に格納された表のを例示する。この例は、第１撹乱用データが０ｘ１ｃ７１ｃ７１ｃ７１ｃ７、第１撹乱用データが０ｘ５５５５５５５５が選択された例である。

【００４３】

図１６は、図１３の実施例で用いる、第１撹乱用データＸ１ｉ（１６０３）および第２撹乱用データＸ２ｉ（１６０４）および変形済み表（１６０５）を作成するための一実施例で、請求項１２の一実施例である。予め第１撹乱用データＸ１ｉに成り得るハミングウエイト一定の値と、第２撹乱用データＸ２ｉに成り得るハミングウエイト一定のデータと、さらにそれらの組を用いて変形処理が行われた変形済み表が複数組が格納された、第１撹乱用データ、第２撹乱用データ及び変形済み表格納手段（１６０２）から、第１撹乱用データ、第２撹乱用データ及び変形済み表選択手段（１６０１）により、第１撹乱用データＸ１ｉ、第２撹乱用データ及び変形済み表の組が１つ選択して取り出され、第１撹乱用データＸ１ｉ（１６０３）、第２撹乱用データＸ２ｉ（１６０４）、変形済み表（１６０５）として選択される。図５８に図１６に対応する第１撹乱用データ、第２撹乱用データ及び変形済み表のテーブルの例を示す。第１撹乱用データ、第２撹乱用データ及び変形済み表格納手段（１６０２）には、このような表を複数セット有している。

【００４４】

図１７は、データ処理がテーブルルックアップとそれに引き続く１つの処理で定義された処理を、２個の撹乱用データを用いて撹乱する実施例で、請求項１３の一実施例である。入力データＤ１（１７０１）を用いて、表を引き、処理ｐを行い、処理済データＤ２（１７１２）得る。表をＴａｂｌｅとすると、Ｄ１とＤ２の関係は、（式２５）のようになる。

Ｄ２＝ｐ（Ｔａｂｌｅ[Ｄ１]）　　（式２７）

この表引き処理を行う際の電流を測定すると、Ｄ１やＤ２の値が推定できる。そこで、第１撹乱用データＸ１ｉと第２撹乱用データＸ２ｉと、表のインデックスと出力結果を変形する関数ｆと関数ｇを考え、変形済みの表ＸＴａｂｌｅを（式２７）のように定義する。

ＸＴａｂｌｅ[ｆ（ｉ、Ｘ１ｉ）]
＝ｇ（Ｔａｂｌｅ[ｉ]、Ｘ２ｉ）　　（式２８）

また、ｇの逆関数をｈと定義する。

Ｄ＝ｈ（ｇ（Ｄ、Ｘ）、Ｘ）　　（式２９）

Ｘ２ｏを次の様に定義する

Ｘ２ｏ＝ｐ（Ｘ２ｉ）　　（式３０）

すると、表引き及び処理ｐは、

Ｈ１＝ｆ（Ｄ１、Ｘ１ｉ）　　（式３１）
Ｈ２＝ＸＴａｂｌｅ[Ｈ１]　　（式３２）
Ｈ３＝ｐ（Ｈ２）　　（式３３）

Ｄ２＝ｈ（Ｈ３、Ｘ２ｏ）　　（式３４）

と表現される。ここで、関数ｆ、関数ｈ、関数ｐは、（式３５）を満たすものである必要がある。

ａ＝ｈ（ｐ（ｆ（ａ、Ｘ））、ｐ（Ｘ））　　（式３５）

この様な条件を満たす関数ｆ、関数ｈ、関数ｐの組み合わせには、

ｆ（ｘ、ｙ）＝ｘ　ｘｏｒ　ｙ　　（式３６）
ｐ（ｘ）　＝右ローテーション（ｘ）
ｈ（ｘ、ｙ）＝ｘ　ｘｏｒ　ｙ

などがある。

【００４５】

ここで、（式３２）の処理を行なう際の消費電流によりＨ１が推定されても、Ｘ１ｉにより撹乱されているため、（式３２）の測定結果だけでは、Ｄ１の値を推定することは出来ない。同様に、（式３３）の処理の消費電流によりＨ３が推定されても、Ｘ２ｉにより撹乱されているため、（式３３）の測定結果だけでは、Ｄ２の値を推定することは出来ない。図１７では、（式３１）にあたる処理は、第１撹乱用データＸ１ｉ（１７０３）を用いて、入力データＤ１（１７０１）をデータ変形処理手段（１７０２）で処理し、変形データＨ１（１７０４）を得る処理に相当する。また、（式３２）に当たる処理は、変形データＨ１（１７０４）をインデックスとして、変形済み表（１７０６）を表引きする変形済み表アクセス手段（１７０５）を用いて、変形データＨ２（１７０７）を得る処理に相当する。（式３３）に相当する処理は、変形データＨ２（１７０７）を変形済みデータ処理手段（１７０８）により処理し、処理済変形データＨ３（１７０９）を得る処理に相当する。（式３４）に相当する処理は、処理済変形データＨ３（１７０９）を処理済第２撹乱用データＸ２ｏ（１７１１）を用いてデータ逆変形

処理手段（１７１０）により逆変形し、処理済データＤ２（１７１２）を得る処理に相当する。

【００４６】

図１９は、図１７の実施例で用いる、第１撹乱用データＸ１ｉ（１９０３）および第２撹乱用データＸ２ｉ（１９０４）および変形済み表（１９０８）および処理済第２撹乱用データＸ２ｏ（１９０９）を作成するための一実施例で、請求項１４の一実施例である。

【００４７】

第１ハミングウエイト一定乱数生成手段（１９０１）により、第１撹乱用データＸ１ｉ（１９０３）を生成し、第２ハミングウエイト一定乱数生成手段（１９０２）により、第２撹乱用データＸ２ｉ（１９０４）が生成される。撹乱用データ処理手段（１９０７）により、第２撹乱用データＸ２ｉ（１９０４）が処理され、処理済第２撹乱用データ（１９０９）が生成される。ハミングウエイト検査手段（１９１０）により、処理済第２撹乱用データ（１９０９）のハミングウエイトが検査され、不適切な場合は、第２ハミングウエイト一定乱数生成手段（１９０２）に対して、再生成制御信号が送られ、第２撹乱用データＸ２ｉ（１９０４）の再生成が行われる。表格納手段（１９０５）に格納された表と、第１撹乱用データＸ１ｉ（１９０３）と、第２撹乱用データＸ２ｉ（１９０４）から、表変形手段（１９０６）により（式２７）を満たす変換が行なわれ、変形済み表（１９０８）を生成する。第１ハミングウエイト一定乱数生成手段（１９０１）および第２撹乱用データＸ２ｉ（１９０４）には、請求項６、請求項７、請求項８のいずれかのハミングウエイト一定乱数生成手段を用いることが出来る。本実施例の利点は、第１撹乱用データ、第２撹乱用データをその都度生成するため、特に撹乱用データのビット長が長い場合に、多くのバリエーションが期待できる点である。

【００４８】

図２０は、図１７の実施例で用いる、第１撹乱用データＸ１ｉ（２００５）および第２撹乱用データＸ２ｉ（２００６）および変形済み表（２０１０）および処理済第２撹乱用データＸ２ｏ（２０１１）を作成するための一実施例で、請求

項１５の一実施例である。

【００４９】

第１撹乱用データ格納手段（２００１）に格納された、複数の第１撹乱用データ候補から、第１撹乱用データ選択手段（２００３）によりデータを選択し、第１撹乱用データＸ１ｉ（２００５）を生成し、第２撹乱用データ格納手段（２００２）に格納された、複数の第２撹乱用データ候補から、第２撹乱用データ選択手段（２００４）によりデータを選択し、第２撹乱用データＸ２ｉ（２００６）を生成する。撹乱用データ処理手段（２００９）により、第２撹乱用データＸ２ｉ（２００６）が処理され、処理済第２撹乱用データ（２０１１）が生成される。表格納手段（２００７）に格納された表と、第１撹乱用データＸ１ｉ（２００５）と、第２撹乱用データＸ２ｉ（２００６）から、表変形手段（２００８）により（式２６）を満たす変換が行なわれ、変形済み表（２０１０）が生成される。本実施例の利点は、予め第１撹乱用データ、第２撹乱用データの項補が用意されているため、第１撹乱用データおよび第２撹乱用データの生成に時間がかからず、かつリーク情報も少なくて済むことである。

【００５０】

図２１は、図１７の実施例で用いる、第１撹乱用データＸ１ｉ（２１０５）および第２撹乱用データＸ２ｉ（２１０６）および変形済み表（２１１０）および処理済第２撹乱用データＸ２ｏ（２１０７）を作成するための一実施例で、請求項１６の一実施例である。

【００５１】

第１撹乱用データ格納手段（２１０１）に格納された、複数の第１撹乱用データ候補から、第１撹乱用データ選択手段（２１０３）によりデータを選択し、第１撹乱用データＸ１ｉ（２１０５）を生成し、第２撹乱用データおよび処理済第２撹乱用データ格納手段（２１０２）に格納された、複数の第２撹乱用データおよび処理済第２撹乱用データ候補の組から、第２撹乱用データおよび処理済第２撹乱用データ選択手段（２１０４）によりデータを選択し、第２撹乱用データＸ２ｉ（２１０６）をおよび処理済第２撹乱用データＸ２ｏ（２１０７）を生成する。表格納手段（２１０９）に格納された表と、第１撹乱用データＸ１ｉ（２１

０５）と、第２撹乱用データＸ２ｉ（２１０６）から、表変形手段（２１０８）により（式２６）を満たす変換が行なわれ、変形済み表（２１１０）が生成される。本実施例の利点は、予め第１撹乱用データ、第２撹乱用データおよび処理済第２撹乱用データの項補が用意されているため、第１撹乱用データ、第２撹乱用データおよび処理済み第２撹乱用データＸ２ｏの生成に時間がかからず、第２撹乱用データから処理済第２撹乱用データを生成する処理がなく、その分請求項１５に比べてさらにリーク情報が少なくて済む。

【００５２】

図２２は、図１７の実施例で用いる、第１撹乱用データＸ１ｉ（２２０３）および処理済第２撹乱用データＸ２ｉ（２１０４）および変形済み表（２２０５）を作成するための一実施例で、請求項１７の一実施例である。

【００５３】

第１撹乱用データと処理済第２撹乱用データと変形済み表の組を複数格納した第１撹乱用データ、処理済第２撹乱用データおよび変換済み表格納手段（２２０２）より、第１撹乱用データ処理済み第２撹乱用データおよび変換済み表選択手段（２２０１）により、第１撹乱用データ処理済み第２撹乱用データおよび変換済み表を選択し、第１撹乱用データＸ１ｉ（２２０３）および処理済第２撹乱用データＸ２ｉ（２１０４）および変形済み表（２２０５）とする。本実施例の利点は、表を変形する処理が不要なため、請求項１６に比べ、さらにリーク情報が少なくて済む。

【００５４】

図２３は、表引きとそれに引き続く１つの処理からなる一連の処理を複数回繰り返す処理装置において、表のインデックスと内容を２つの撹乱用データを用いて処理中に現れる数値を撹乱し、処理を行なう情報処理装置の１実施例で、請求項１８の１実施例である。

【００５５】

複数回繰り返して処理を行なう際に、変形されているデータを逆変形する際に、まず次の処理のための変形を行なった後に、逆変形を行なうという手順を取るため、変形されていないデータが途中では現れず、途中では、２重に変形されて

いるか、１重に変形されているかの違いはあるが、常に途中のデータは変形されたまま処理されるため、リーク情報が少ないのが特徴である。

【００５６】

この請求項１８で用いられるデータ変形およびデータ逆変形手段は、実行する順番を変えても同じ値を返す必要がある。データｘを撹乱情報ｙ変形する処理を関数ｆ（ｘ、ｙ）とし、変形データｘを撹乱情報ｙで逆変形する関数をｇ（ｘ、ｙ）とすると、

ｆ（ｇ（ｘ、ｙ１）、ｙ２）＝ｇ（ｆ（ｘ、ｙ２）、ｙ１）　（式３７）

なる条件を満たす必要がある。

【００５７】

実施例中の、第１撹乱用データＸ１ｉ（２３０３）、変形済み表（２３０６）、変形済み第２撹乱用データ（２３１３）は、請求項１４、請求項１５、請求項１６、請求項１７の実施例のいずれかにより生成することが可能である。入力データＤ１（２３０１）を第１撹乱用データＸ１ｉを用いてデータ変形処理手段（２３０２）で変形し、変形データＨ１（２３０４）を得る。変形データＨ１（２３０４）を用いて、変形済み表（２３０６）を変形済み表アクセス処理手段（２３０５）により表引き処理を行い、変形データＨ２（２３０７）を得る。変形済みデータ処理手段（２３０８）にて、変形データＨ２（２３０７）を処理し、処理済変形データＨ３（２３０９）を得る。この状態で、Ｈ２は、第２撹乱用データによる変形を受けた状態にある。さらに、第１撹乱用データＸ１ｉ（２３０３）を用いて、処理済変形データＨ３（２３０９）を、データ変形処理手段（２３１０）で変形し、処理済変形データＨ４（２３１１）を得る。Ｈ４は、第１撹乱用データと第２撹乱用データの２つにより変形された状態にある。ここで、変形済み第２撹乱用データＸ２ｏ（２３１３）を用いて、データ逆変形処理手段（２３１２）により、処理済変形データＨ５（２３１４）を得る。Ｈ５は、第２撹乱用データによる変形が解かれ、第１撹乱用データのみによって変形している状態となり、変形済み表のインデックスとして使用可能である。処理済変形データＨ

５（２３１４）を用いて、変形済み表アクセス処理手段（２３１５）により、変形済み表（２３０６）の表引きを行い、処理済変形データＨ６（２３１６）を得る。処理済変形データＨ６（２３１６）を変形済みデータ処理手段（2317）で処理し、処理済変形データＨ７（２３１８）を得る。Ｈ７は、第２撹乱データで変形を受けているため、最終結果とするために、変形済み第２撹乱データＸ２ｏ（２３１３）を用いて、逆変形を行い、最終的な結果である、処理済データＤ２（２３２０）を得る。この実施例は、一連の処理の繰り返し回数は、２回のみであるが、同様の手順で複数回の繰り返しが可能である。

【００５８】

図１９は、図２３の実施例で用いる、第１撹乱用データＸ１ｉ（１９０３）および変形済み表（１９０８）および処理済第２撹乱用データＸ２ｏ（１９０９）を作成するための一実施例でもあり、図１９と図２３をあわせた実施例は、請求項１９の一実施例である。

【００５９】

図２０は、図２３の実施例で用いる、第１撹乱用データＸ１ｉ（２００５）および変形済み表（２０１０）および処理済第２撹乱用データＸ２ｏ（２０１１）を作成するための一実施例でもあり、図２０と図２３をあわせた実施例は、請求項２０の一実施例である。

【００６０】

図２１は、図２３の実施例で用いる、第１撹乱用データＸ１ｉ（２１０５）および変形済み表（２１１０）および処理済第２撹乱用データＸ２ｏ（２１０７）を作成するための一実施例でもあり、図２１と図２３をあわせた実施例は、請求項２１の一実施例である。

【００６１】

図２２は、図２３の実施例で用いる、第１撹乱用データＸ１ｉ（２２０３）および変形済み表（２２０４）および処理済第２撹乱用データＸ２ｏ（２２０５）を作成するための一実施例でもあり、図２２と図２３をあわせた実施例は、請求項２２の一実施例である。

【００６２】

図２４は、表引きとそれに引き続く１つの処理からなる一連の処理を複数回繰り返す処理装置において、表のインデックスと出力をそれぞれ２つの撹乱用データで２重に変形し、処理する実施例の一つで、請求項２３の実施例の一つである。２重に変形することにより、消費電流による観測を少ない資源でより強力な撹乱が行なえるようにする。

【００６３】

この実施例で用いられる、４つの撹乱用データと、前記撹乱用データを用いて第２変形済み表を生成する方法としては、請求項１９、２０、２１、２２がある。たとえば、予めハミングウエイト一定の数値を複数個用意して選択する実施例の場合、予め用意した数値のバリエーションが少ない場合、撹乱データを用いてデータを変形する処理が既知の場合、予め用意されている撹乱用データを全て推定することも可能となる。撹乱用データを用いて変形する処理がXOR処理であった場合、元のデータと撹乱用データが同一のデータであった場合、変形済みのデータは０となることから、予め用意された撹乱用データのセットを推定することが不可能ではない。そこで、たとえばハミングウエイトを固定しない代りに常にビット長で表現できる全ての値を生成した撹乱用データで変形した後に、ハミングウエイト一定の撹乱用データを用いてさらに変形を行なうことで、容易には撹乱用データの推定ができなくなる。図２４を用いて、処理の詳細を説明する。

【００６４】

入力データＤ１（２４０１）が、第３撹乱用データＸ３ｉ（２４０３）を用いて、データ変形手段（２４０２）で変形され、変形データＨ１（２４０４）が生成される。第３撹乱用データは、表引き処理に用いられる表のインデックスを変形する２つの撹乱データのうちの一つで、実際の表引きには、さらに第１撹乱用データＸ１ｉ（２４０６）で変形する必要がある。その変形処理を行なうのが、データ変形処理手段（２４０５）で、変形データＨ１（２４０４）を第１撹乱用データ（２４０６）を用いて変形し、変形データＨ２（２４０７）を得る。得られた変形データＨ２（２４０７）をインデックスとして用い、変形済み表アクセス処理手段（２４０８）により第２変形済み表（２４０９）を表引きし、変形データＨ３（２４１０）を得る。変形データＨ３（２４１０）を処理する変形済み

データ処理手段（２４１１）で処理し、処理済変形データＨ４（２４１２）を生成する。第３撹乱用データＸ３ｉ（２４０３）を用いて、データ変形処理手段（２４１３）により、処理済変形データＨ４（２４１２）を変形し、処理変形データＨ５（２４１４）を得て、さらに第１撹乱用データＸ１ｉ（２４０６）を用いてデータ変形処理手段（２４１５）で変形し、処理済変形データＨ６（２４１６）を得る。ここまでで、処理中のデータは、第３撹乱用データＸ３ｉ（２４０３）、第１撹乱用データＸ１ｉ（２４０６）、および第２変形表（２４０９）に畳み込まれている、処理済第２撹乱用データＸ２ｏ（２４１８）、処理済第４撹乱用データＸ４ｏ（２４２１）で変形されている状態なので、まずデータ逆変換手順（２４１７）で、処理済第２撹乱用データＸ２ｏ（２４１８）を用いて、データの逆変形を行い、処理済変形データＨ７（２４１９）を生成する。そしてデータ逆変換手順（２４２０）で、処理済第４撹乱用データＸ４ｏ（２４２１）を用いて、データの逆変形を行い、処理済変形データＨ８（２４２２）を生成する。これで、データは第３撹乱用データＸ３ｉ（２４０３）、第１撹乱用データＸ１ｉ（２４０６）の２つの撹乱用データで変形されている状態になるので、第２変形済み表（２４０９）のインデックスとして用いることが出来る。変形済み表アクセス処理手順（２４２３）で表引きを行い、変形データＨ９（２４２４）を得て、さらに変形済みデータ処理手順（２４２５）により、処理済変形データＨ１０（２４２６）を得る。ここまでで、最終結果を処理済第２撹乱用データＸ２ｏ（２４１８）、処理済第４撹乱用データＸ４ｏ（２４２１）で変形されている状態のデータが得られたことになるので、データ逆変換処理手順（２４２７）により、処理済第２撹乱用データＸ２ｏ（２４１８）による変形分を逆変形し、さらにデータ逆変換処理手順（２４２９）により、処理済第４撹乱用データＸ４ｏ（２４２１）による変形分を逆変形して最終的な結果Ｄ２（２５３０）を得る。

【００６５】

図２６は、図２４すなわち請求項２３の実施例における、第１撹乱用データＸ１ｉ（２６０２）、処理済第２撹乱用データＸ２ｏ（２６０６）、第３撹乱用データＸ３ｉ（２６１２）、処理済第４撹乱用データＸ４ｏ（２６１８）と、第２処理済表（２６１７）を生成する方法の１実施例であり、請求項２４の一実施例

である。処理手順は、まず図１９の実施例の方法で、第１撹乱用データＸ１ｉ（２６０２）、処理済第２撹乱用データＸ２ｏ（２６０６）、および変形済み表（２６１０）を生成し、生成した変形済み表（２６１０）をさらに第３撹乱用データ生成手段（２６１１）で生成された第３撹乱用データＸ３ｉ（２６１２）と、第４撹乱用データ生成手段（２６１３）で生成された第４撹乱用データＸ４ｉ（２６１４）とを用いて、表変形処理手段（２６１６）にて変形済み表（２６１０）をさらに変形して、第２変形済み表（２６１７）を得る。また、データの逆変換時に必要となる、処理済み第４撹乱用データＸ４ｏ（２６１８）を第４撹乱用データＸ４ｉ（２６１４）から、撹乱用データ処理手段（２６１５）を用いて計算しておく。

【００６６】

図２７は、図２４すなわち請求項２３の実施例における、第１撹乱用データＸ１ｉ（２７０３）、処理済第２撹乱用データＸ２ｏ（２７０７）、第３撹乱用データＸ３ｉ（２７１２）、処理済第４撹乱用データＸ４ｏ（２７１８）と、第２処理済表（２７１４）を生成する別の方法の位置実施例であり請求項２５の一実施例である。

【００６７】

処理手順は、まず図２０の実施例の方法で、第１撹乱用データＸ１ｉ（２７０３）、処理済第２撹乱用データＸ２ｏ（２７０６）、および変形済み表（２７１０）を生成し、生成した変形済み表（２７１０）をさらに第３撹乱用データ生成手段（２７１１）で生成された第３撹乱用データＸ３ｉ（２７１２）と、第４撹乱用データ生成手段（２７１５）で生成された第４撹乱用データＸ４ｉ（２７１６）とを用いて、表変形処理手段（２７１３）にて変形済み表（２７１０）をさらに変形して、第２変形済み表（２７１４）を得る。また、データの逆変換時に必要となる、処理済み第４撹乱用データＸ４ｏ（２７１８）を第４撹乱用データＸ４ｉ（２６１６）より、撹乱用データ処理手段（２７１７）を用いて計算しておく。

【００６８】

図２８は、図２４すなわち請求項２３の実施例における、第１撹乱用データＸ

１ｉ（２９０４）、処理済第２撹乱用データＸ２ｏ（２８０５）、第３撹乱用データＸ３ｉ（２８０７）、処理済第４撹乱用データＸ４ｏ（２８０４）と、第２処理済表（２９０９）を生成する別の方法の位置実施例であり請求項２６の一実施例である。

【００６９】

処理手順は、まず図２１の実施例の方法で、第１撹乱用データＸ１ｉ（２８０４）、処理済第２撹乱用データＸ２ｏ（２８０５）、および変形済み表（２８０３）を生成し、生成した変形済み表（２８０３）をさらに第３撹乱用データ生成手段（８０６）で生成された第３撹乱用データＸ３ｉ（２８０７）と、第４撹乱用データ生成手段（２８１０）で生成された第４撹乱用データＸ４ｉ（２８１１）とを用いて、表変形処理手段（２７１３）にて変形済み表（２８０３）をさらに変形して、第２変形済み表（２８０９）を得る。また、データの逆変換時に必要となる、処理済み第４撹乱用データＸ４ｏ（２８１３）を第４撹乱用データＸ４ｉ（２８１１）より、撹乱用データ処理手段（２８１２）を用いて計算しておく。

【００７０】

図２９、３０、３１、３２、３３、３４、３５、３６、３７、３８、３９、４５、４６、４７を用いて、請求項２７の一実施例について説明する。

【００７１】

まず、図２９を用いて、ＳＢＯＸ表の変形処理および撹乱用データについて説明する。ＳＢＯＸ表のアドレス撹乱用データＳｉｎＸ１（２９０１）とＳＢＯＸデータ撹乱用データＳｏｕｔＸ（２９０２）を用いて、ＳＢＯＸ表（２９０３）をＳＢＯＸ表変形手段（２９０４）で変形処理を行い、変形済みＳＢＯＸ表（２９０５）を得る。ＳＢＯＸ表のアドレスおよびデータの変形は、ＸＯＲで行なうものとする。また、ＳｏｕｔＸを転置処理Ｐ（２９０６）、転置処理Ｅ（２９０７）を行い、転置済みＳＢＯＸデータ撹乱用データＸＳｏｕｔＸ（２９０９）を生成しておく。ここでの処理を纏めると、ＳＢＯＸ表をＳＢＯＸ[0..63]、変形ＳＢＯＸ表をXSBOX[0..63]、転置処理Ｐを関数Ｐ（）、拡大転置処理Ｅを関数Ｅ（）で表現すると、

ＸＳＢＯＸ［ｉ xor SinX1]=SBOX[i] xor SoutX　　（式３８）

ＸＳｏｕｔＸ＝Ｅ（Ｐ（ＳｏｕｔＸ））　　（式３９）

となる。ＳｉｎＸ１、ＳｏｕｔＸの作成方法は、請求項１４、１５、１６、１７のいずれかの方法を用いることができる。

【００７２】

図４６は、ＳＢＯＸの格納様式の一実施例である。６４個の３２ビット整数からならる要素を持つ１次元の配列としてＳＢＯＸを格納する。図４５は、図４６の様式で格納されたＳＢＯＸを表引きするための処理の一実施例で、４８ビットの入力を６ビットごとに分解し、ＳＢＯＸ配列を表引きし、表引きの際に用いた６ビットの値の位置に応じたマスクで必要なデータを取り出し、順次加算することで、最終的な表引き結果を得る。図４５のフローチャートを用いて、処理を順を追って説明する。４５０２で、ｉｎに表引きの入力となる４８ビットの数値を代入する。４８ビットを６ビットづつに分解し、８回に分けて処理するため、４５０３において、処理回数をカウントするための変数ｊを０に初期化する。４５０４で、表引き結果をマスクするための変数maskを１５で初期化する。これは、下位４ビットがすべて１である数値である。４５０５で、表引き結果を格納する変数ｒｅｓｕｌｔを０に初期化する。４５０６で、ｉｎの下位６ビットを取り出し、ｉｄｘに代入する。４５０７で、ｉｎを右に６ビットシフトし、つぎの６ビットを取り出す準備を行う。４５０８でｄに図４６のＳＢＯＸをｉｄｘをインデックスとしたときの表引き結果を代入する。４５０９でｄとｍａｓｋのＡＮＤを計算し、ｄに代入する。４５１０で、ｒｅｓｕｌｔにｄを加える。４５１１にて、ｍａｓｋを左に４ビットシフトし、次のデータをマスクする準備とする。４５１２でカウンタｊを１だけ加算する。４５１３でｊが８未満であれば、４５０６からの処理を繰り返す。４５１４では、ｒｅｓｕｌｔを結果として返す。

【００７３】

図４７は、図２９のＳＢＯＸ表変形手段（２３０４）の詳細なフローチャートであり、図４６の様式で格納されたＳＢＯＸを変形するための手順を示す。この

手順で変形ささされた変形済みＳＢＯＸ表は、図４６の表引き処理により、表引きが可能である。また、図４７で選られた変形済みＳＢＯＸを通常のＳＢＯＸと見なして、新たな撹乱用データを用いて図４７の処理を繰り返すことで、多重に変形することが可能である。図４７にしたがって、ＳＢＯＸ変形の手順を説明する。４７０２では、ＳＢＯＸを表引きするためのインデックスidxを０に初期化する。４７０３では、ｉｄｘを６ビットのビット列とみなし、そのビット列を８個繰り返して選られる４８ビットの数値を計算し、ｉｎに代入する。４７０３では、ｉｎと、４８ビットアドレス撹乱用データとのｘｏｒを計算し、ｉｎに代入する。４７０５では、ｉｎをＳＢＯＸの表引き用の４８ビットの入力として、図４５に示されるＳＢＸＯ表アクセス手順を呼び出し、結果をｒｅｓｕｌｔとする。４７０６では、結果ｒｅｓｕｌｔに対して、３２ビットの出力撹乱用データとのｘｏｒを計算し、ｒｅｓｕｌｔに代入する。４７０７では、変形済みＳＢＯＸ表のインデックスｉｄｘの位置に、ｒｅｓｕｌｔを代入する。４７０８では、ｉｄｘに１を加える。４８０９では、ｉｄｘが６４未満かどうかを判定し、６４未満の場合は、４７０３からの処理を繰り返す。ｉｄｘが６４になると、表の変換が終了する。

【００７４】

次に、図３０を用いて、平文撹乱用データＰＸで変形したデータを、逆変換するために用いる、置換済み平文撹乱用データＰＸｏ１（3003）、ＰＸｏ２（３００７）、ＰＸｏ３（3006）、ＰＸｏ４（3010）の生成について説明する。平文用データＰＸを転置処理ＩＰ（３００２）で転置し、上位３２ビットと下位３２ビットにわけ、それぞれ転置済み平文撹乱用データ１のＰＸｏ１（３００３）、転置済み平文撹乱用データ２のＰＸｏ２（３００７）とする。この２つの値は、最終ラウンドの処理が終わった後の、ＩＰ逆転置処理の直前で変形データを最終的な結果にするための逆変形処理に用いられる。つぎに、転置済み平文撹乱用データ１のＰＸｏ１（３００３）、転置済み平文撹乱用データ２のＰＸｏ２（３００７）に対して、拡大転置処理Ｅ（３００５、３００９）を行ない、転置済み平文撹乱用データ３のＰＸｏ３（３００６）、転置済み平文撹乱用データ４のＰＸｏ４（３０１０）とし、各ラウンドでの変形ＳＢＯＸ表の表引き前の逆変形処理に

用いる。

【００７５】

次に、図３１を用いて、秘密鍵撹乱用データと、各ラウンドごとの鍵の処理のうち、ＬＳ処理の直後に行なう変形処理用に用いる、処理済み秘密鍵撹乱用データＫＸｏ１（３１０９）、ＫＸｏ２（３１１１）、ＫＸｏ３（３１１３）について、説明する。本実施例では、選択転置処理ＰＣ２での通常の出力を、Ｘとした場合、

Ｘ　ｘｏｒ　ＸＳｏｕｔＸ　　（式４０）

という値を出力するようにしたい。選択置換ＰＣ１をＰＣ１（）と表現し、ＬＳ処理をＬＳ（）と表現し、鍵をＫと表現する。本実施例では、鍵は秘密鍵撹乱用データＫＸとのｘｏｒで撹乱されるため、第１ラウンドでは、

Ｋ０＝ＬＳ（ＰＣ１（ＫＸ　ｘｏｒ　Ｋ））　　（式４１）

ＫＸｏ１＝ＬＳ（ＰＣ１（ＫＸ））　ｘｏｒ
ＩＮＶ＿ＰＣ２（ＳｉｎＸ１）　　（式４２）

Ｋ１＝ＫＯ　ｘｏｒ　ＫＸｏ１　　（式４３）
Ｋ１＿ＯＵＴ＝ＰＣ２（Ｋ１）　　（式４４）

Ｋ１＿ＯＵＴをＰＣ２からの出力として使うことで、（式４０）を満たす値が得られる。つぎに、第２ラウンド用の値は、

ＫＸｏ２＝ＩＮＶ＿ＰＣ２（ＳｉｎＸ１）ｘｏｒ
ＬＳ（ＩＮＶ＿ＰＣ２（ＳｉｎＸ１））　（式４５）

Ｋ２＝ＬＳ（Ｋ１）　ｘｏｒ　ＫＸｏ２　　（式４６）

Ｋ２＿ＯＵＴ＝ＰＣ２（Ｋ２）

となる。また、たとえばラウンド３のように、ＬＳ処理で２ビット分ローテートするラウンドでは、

ＫＸｏ３＝ＩＮＶ＿ＰＣ２（ＳｉｎＸ１）ｘｏｒ
ＬＳ（ＬＳ（ＩＮＶ＿ＰＣ２（ＳｉｎＸ１））） （式４７）
Ｋ３＝ＬＳ（ＬＳ（Ｋ２））ｘｏｒ ＫＸｏ３
Ｋ３＿ＯＵＴ＝ＰＣ２（Ｋ３） （式４８）

とすることで、（式４０）を満たす値が得られる。ＬＳ処理でシフトするビットの種類は２種類のみなので、ラウンド１用のＫＸｏ１と、ＬＳ処理のシフト量が１ビットのＫＸｏ２、ＬＳ処理のシフト量が２ビットのＫＸｏ３の３種類の値が必要となる。これらの３つの値で、ラウンド１６全ての変形処理を行なうことが出来る。（式４１）、（式４５）、（式４７）で示されるＫＸｏ１、ＫＸｏ２、ＫＸｏ３の計算は、図３１のシグナルフローの通りである。本実施例では、変形処理にｘｏｒ演算を用いるので、図３１中の、逆変換結合処理（３１０８、３１１０、３１１２）はｘｏｒ処理を用いる。

【００７６】

図３２は、平文データＰｔｅｘｔ（３２０１）を変形する処理の一実施例である。

【００７７】

平文撹乱用データＰＸ（３２０３）を用いて、第１変形処理（３２０２）で変形し、変形済み平文データＸＰｔｅｘｔ（３２０４）を作成する。ここで、第１変形処理（３２０２）は、本実施例では、ｘｏｒ演算を用いる。したがって、

ＸＰｔｅｘｔ＝Ｐｔｅｘｔ ｘｏｒ ＰＸ （式４９）

となる。ＸＰｔｅｘｔを転置処理ＩＰ（３２０５）により転置し、上位３２ビッ

トを転置済み変形平文１のＸＰｔｅｘｔＬ（３２０６）、下位３２ビットを転置済み変形平文２のＸＰｔｅｘｔＲ（３２０７）とする。この処理は、第１変形処理（３２０２）を除けば、通常のＤＥＳ暗号処理の処理フローに等しい。

【００７８】

ここで、別の実施例を図５４に示す。この別の実施例では、平文データ（５４０１）を転置処理ＩＰ（５４０２）で処理した後に、第１変形手段（５４０４、５４０７）で撹乱している。この実施例では、平文撹乱用データＰＸ（５３０１）の処理が、転置処理ＩＰを行わずに済む分だけ、効率的である。図５３は、平文撹乱用データの処理の１実施例である。

【００７９】

図３３は、秘密鍵データＫ（３３０１）の変形処理の一実施例で、秘密鍵撹乱用データＫＸ（３３０３）を用いて、第２変形処理（３３０２）にて、秘密鍵データＫ（３３０１）を変形し、変形済み秘密鍵データＸＫ（３３０４）を得る。本実施例では、第２変形処理手段として、ｘｏｒ演算を用いる。したがって、

ＸＫ＝Ｋ　ｘｏｒ　ＫＸ　　（式５０）

となる。つぎに各ラウンドの処理を図３４、図３５、図３６、図３７、図３８に示す。

【００８０】

５つの図の違いは、ラウンドの違いで、第１ラウンド、第５ラウンド、第９ラウンド、第１３ラウンドの処理が図３５、第２ラウンド、第６ラウンド、第１０ラウンド、第１４ラウンドの処理が、図３５、第３ラウンド、第７ラウンド、第１１ラウンド、第１５ラウンドの処理が、図３６、第４ラウンド、第８ラウンド、第１２ラウンドの処理が図３７、第１６ラウンドの処理が図３８に示されている。

【００８１】

図３４の処理を順を追って説明する。まず、撹乱処理を行わないときのＸＰｔｅｘｔＬ（３４０１）に来るべき値をＰｔｅｘｔＬ、同様に撹乱処理を行わなか

ったときのＸＰｔｅｘｔＲ（３４０２）に来るべき値をＰｔｅｘｔＬとする。すると、ＸＰｔｅｘｔＬおよびＸＰｔｅｘｔＲは、

ＸＰｔｅｘｔＬ＝ＰＴｅｘｔＬ　ｘｏｒ　ＰＸｏ１　　（式５１）
ＸＰｔｅｘｔＲ＝ＰＴｅｘｔＲ　ｘｏｒ　ＰＸｏ２　　（式５２）

と表現される。同様に、ＸＫＬ（３４０７）の値をＸＫＬ０、撹乱処理を行わない場合のＸＫＬ（３４０７）に来るべき値をＫＬ、選択置換処理ＰＣ１を関数ＰＣ１（）として表現すると、

ＸＫＬ０＝ＫＬ　ｘｏｒ　ＰＣ１（ＫＸ）　　（式５３）

となる。第３変形処理（３４０９）を実行後のＸＫＬ（３４１０）の値をＸＫＬ１とし、選択置換ＰＣ２の逆関数をＩＮＶ＿ＰＣ２（）とする。ここで、ＰＣ２（）で参照されないビットに付いては、ＩＮＶ＿ＰＣ２（）では０になるものとする。第３逆変形処理（３４０９）で使用される処理済秘密鍵撹乱用データは、そのラウンドで行われるローテート処理ＬＳ（３４０８）でシフトされるビット数で決まり、１ビットの場合は、ＫＸｏ２、２ビットの場合は、ＫＸｏ３、ラウンド１の場合は、ＫＸｏ１が用いられる。

ＸＫＬ１＝ＬＳ（ＸＫＬ０）　ｘｏｒ　ＫＸｏ１　　（式５４）

さらに、（式５３）と（式４２）を代入すると、

ＸＫＬ１
＝ＬＳ（ＫＬ　ｘｏｒ　ＰＣ１（ＫＸ））ｘｏｒ　ＫＸｏ１　（式５５）

また、

ＬＳ（ａ　ｘｏｒ　ｂ）＝ＬＳ（ａ）　ｘｏｒ　ＬＳ（ｂ）　（式５６）

（ａ　ｘｏｒ　ｂ）　ｘｏｒ　ｃ
＝ａ　ｘｏｒ　（ｂ　ｘｏｒ　ｃ）　　（式５７）

という関係を用いて、（式５４）を書き直すと、

ＸＫＬ１＝ＬＳ（ＫＬ　ｘｏｒ　ＰＣ１（ＫＸ））ｘｏｒ
（ＬＳ（ＰＣ１（ＫＸ））ｘｏｒ
ＩＮＶ＿ＰＣ２（ＳｉｎＸ１））
＝ＬＳ（ＫＬ　ｘｏｒ　ＰＣ１（ＫＸ）　ｘｏｒ
ＰＣ１（ＫＸ））　ｘｏｒ　ＩＮＶ＿ＰＣ２（ＳｉｎＸ１）
＝ＬＳ（ＫＬ）　ｘｏｒ　ＩＮＶ＿ＰＣ２（ＳｉｎＸ１）　（式５８）

となる。この値に選択置換処理ＰＣ－２（３４１４）を施した値をＸＫＬ１ＰＣ２とすると、
ＸＫＬ１ＰＣ２＝ＰＣ２（ＸＫＬ１）
＝ＰＣ２（ＬＳ（ＫＬ）　ｘｏｒ　ＩＮＶ＿ＰＣ２（ＳｉｎＸ１））
＝ＰＣ２（ＬＳ（ＫＬ））　ｘｏｒ　ＳｉｎＸ１　　（式５９）

となる。ここでは、ラウンド１の場合について説明したが、ラウンド５、ラウンド９、ラウンド１３においても、選択置換処理ＰＣ－２（３４１４）の出力は、撹乱処理を行わない場合の値である　ＰＣ２（ＬＳ（ＫＬ））とＳｉｎＸ１のｘｏｒを行った値となる。

【００８２】

つぎにｘｏｒ演算（３４０４）を行った結果を、ＸＰｔｅｘｔＲＸとし、拡大転置処理Ｅ（３４０３）の演算を関数Ｅ（）と表現すると、ＸＰｔｅｘｔＲＸは

ＸＰｔｅｘｔＲＸ
＝Ｅ（ＸＰｔｅｘｔＲ）　ｘｏｒ　ＸＫＬＰＣ２　　（式６０）

となり、（式５２）、（式５９）を代入すると、

ＸＰｔｅｘｔＲＸ＝Ｅ（ＰｔｅｘｔＲ　ｘｏｒ　ＰＸｏ２）　ｘｏｒ
ＰＣ２（ＬＳ（ＫＬ））　ｘｏｒ　ＳｉｎＸ１
＝Ｅ（ＰｔｅｘｔＲ）ｘｏｒ　ＰＣ２（ＬＳ（ＫＬ））　ｘｏｒ
Ｅ（ＰＸｏ２）　ｘｏｒ　ＳｉｎＸ１　　（式６１）

となる。転置済み平文撹乱用データ４のＰＸｏ４（３４１６）を用いて、第１逆変形処理（３４１５）で逆変形した結果を、ＸＰｔｅｘｔＲＸ２とすると、第１逆変形処理はｘｏｒ演算であるので、

ＸＰｔｅｘｔＲＸ２＝ＸＰｔｅｘｔＲ　ｘｏｒ　ＰＸｏ４
＝Ｅ（ＰｔｅｘｔＲ）　ｘｏｒ　ＰＣ２（ＬＳ（ＫＬ））
ｘｏｒ　Ｅ（ＰＸｏ２）　ｘｏｒ　ＳｉｎＸ１
ｘｏｒ　ＰＸｏ４　　（式６２）

また、図３０の実施例より、

ＰＸｏ４＝Ｅ（ＰＸｏ２）　　（式６３）

であるので、（式６２）は、

ＸＰｔｅｘｔＲＸ２＝Ｅ（ＰｔｅｘｔＲ）　ｘｏｒ　ＰＣ２（ＬＳ（ＫＬ））
ｘｏｒ　Ｅ（ＰＸｏ２）　ｘｏｒ　ＳｉｎＸ１
ｘｏｒ　Ｅ（ＰＸｏ２）
＝Ｅ（ＰｔｅｘｔＲ）　ｘｏｒ　ＰＣ２（ＬＳ（ＫＬ））

ｘｏｒ　ＳｉｎＸ１　　（式６４）

となる。また、撹乱処理を行わない場合の、変形ＳＢＯＸアクセス処理（３４１８）の入力となる値ＰｔｅｘｔＲＸ２は、

ＰｔｅｘｔＲＸ２
＝Ｅ（ＰｔｅｘｔＲ）　ｘｏｒ　ＰＣ２（ＬＳ（ＫＬ））　　（式６５）

であり、（式６４）は、

ＸＰｔｅｘｔＲＸ２
＝ＰｔｅｘｔＲＸ２　ｘｏｒ　ＳｉｎＸ１　　（式６６）

となる。撹乱処理を行わない値と比較すると、ＳＢＯＸのアドレス撹乱用データＳｉｎＸ１とｘｏｒを取った値と等しくなるため、変形済みＳＢＯＸ表（３４１９）をアクセス屡事ができる。アクセスした結果には、ＳｏｕｔＸによる変形が行われているので、ｘｏｒ処理（３４２１）への入力は、撹乱を行わない場合のＳＢＯＸの出力をＳＲｅｓｕｌｔとすると、

Ｐ（ＳＲｅｓｕｌｔ　ｘｏｒ　ＳｏｕｔＸ　　（式６７）

となる。（式６７）と転置済み変形平文１のＸＰｔｅｘｔＬ（３４０１）とのｘｏｒ（３４２１）を計算すると、

Ｐ（Ｓｒｅｓｕｌｔ　ｘｏｒ　ＳｏｕｔＸ）　ｘｏｒ　ＸＰｔｅｘｔＬ
＝Ｐ（Ｓｒｅｓｕｌｔ）ｘｏｒ　Ｐ（ＳｏｕｔＸ）　ｘｏｒ　ＰｔｅｘｔＬ　ｘｏｒ
ＰＴｅｘｔＬ　ｘｏｒ　ＰＸｏ１　　（式６８）

となる。この値は、転置済み変形平文２のＸＰｔｅｘｔＲ（３４２３）に代入されるが、ここで、撹乱を行わない場合のＸＰｔｅｘｔＲ（３４２３）に代入される値を、ＰｔｅｘｔＲ２、撹乱を行った場合の値をＸＰｔｅｘｔＲ２とすると、

ＸＰｔｅｘｔＲ２＝ＰｔｅｘｔＲ２　ｘｏｒ　Ｐ（ＳｏｕｔＸ）
ｘｏｒ　ＰＸｏ１　　（式６９）

同様にして、撹乱を行わない場合のＸＰｔｅｘｔＬ（３４２２）に代入される値を、ＰｔｅｘｔＬ２、撹乱を行った場合の値をＸＰｔｅｘｔＬ２とすると、

ＸＰｔｅｘｔＬ２＝ＰｔｅｘｔＬ２　ｘｏｒ　ＰＸｏ２　（式７０）

（式６９）、（式７０）の値は、次のラウンドの図３５で使用される。（式６９）と（式５１）を比較した場合、ＰＸｏ２の代わりにＰＸｏ１が使われ、さらにＰ（ＳｏｕｔＸ）でのｘｏｒが加わっている。この違いにより、図３４と図３５では、第１逆変形処理（３５１５）に用いられる転置済み平文撹乱用データ（３５１６）が、ＰＸｏ４からＰＸｏ３に替わり、Ｐ（ＳｏｕｔＸ）による変形を元に戻すために、第４逆変形処理（３５１７）の処理が加わっている。第４逆変形処理（３５１７）にたどり着くまでに、Ｐ（ＳｏｕｔＸ）は、拡大置換処理Ｅ（）が施され、Ｅ（Ｐ（ＳｏｕｔＸ））となり、転置済みＳＢＯＸ撹乱用データＸＳｏｕｔＸと等しくなる。

【００８３】

図３５の撹乱を行わない場合のＸＰｔｅｘｔＲ（３５２５）に代入される値を、ＰｔｅｘｔＲ３、撹乱を行った場合の値をＸＰｔｅｘｔＲ３とし、、撹乱を行わない場合のＸＰｔｅｘｔＬ（３５２４）に代入される値を、ＰｔｅｘｔＬ３、撹乱を行った場合の値をＸＰｔｅｘｔＬ３とすると、

ＸＰｔｅｘｔＲ３＝ＰｔｅｘｔＲ３　ｘｏｒ　Ｐ（ＳｏｕｔＸ）
ｘｏｒ　ＰＸｏ２　　（式７１）

ＸＰｔｅｘｔＬ３＝ＰｔｅｘｔＬ３　ｘｏｒ　Ｐ（ＳｏｕｔＸ）
ｘｏｒ　ＰＸｏ１　　（式７２）

となる。

【００８４】

（式７１）、（式７２）の値は、次のラウンドの図３６で使用される。（式７１）と（式６９）を比較した場合、ＰＸｏ１の代わりにＰＸｏ２が使われている。この違いにより、図３５と図３６では、第１逆変形処理（３６１５）に用いられる転置済み平文撹乱用データ（３５１６）が、ＰＸｏ３からＰＸｏ４に替わっている。また、ｘｏｒ処理（３６２３）の２つの入力のいずれにも、Ｐ（ＳｏｕｔＸ）がｘｏｒされているため、Ｐ（ＳｏｕｔＸ）の影響が消える。その結果、図３６の撹乱を行わない場合のＸＰｔｅｘｔＲ（３６２５）に代入される値を、ＰｔｅｘｔＲ４、撹乱を行った場合の値をＸＰｔｅｘｔＲ４とし、撹乱を行わない場合のＸＰｔｅｘｔＬ（３６２４）に代入される値を、ＰｔｅｘｔＬ４、撹乱を行った場合の値をＸＰｔｅｘｔＬ４とすると、

ＸＰｔｅｘｔＲ４＝ＰｔｅｘｔＲ４　ｘｏｒ　ＰＸｏ１　（式７３）
ＸＰｔｅｘｔＬ４＝ＰｔｅｘｔＬ４　ｘｏｒ　Ｐ（ＳｏｕｔＸ）
ｘｏｒ　ＰＸｏ２　　（式７４）

となる。

【００８５】

（式７３）、（式７４）の値は、次のラウンドの図３７で使用される。（式７３）と（式７１）を比較した場合、ＰＸｏ２の代わりにＰＸｏ１が使われており、またＰ（ＳｏｕｔＸ）によるｘｏｒが含まれていない。この違いにより、図３６と図３７では、第１逆変形処理（３７１５）に用いられる転置済み平文撹乱用データ（３５１６）が、ＰＸｏ４からＰＸｏ３に替わっている。Ｐ（ＳｏｕｔＸ）による変形を元に戻す必要が無いため、第４逆変形処理が無くなっている。また、ｘｏｒ処理（３７２１）の２つの入力のいずれにも、Ｐ（ＳｏｕｔＸ）がｘ

ｏｒされているため、Ｐ（ＳｏｕｔＸ）の影響が消える。その結果、図３７の撹乱を行わない場合のＸＰｔｅｘｔＲ（３７２２）に代入される値を、ＰｔｅｘｔＲ５、撹乱を行った場合の値をＸＰｔｅｘｔＲ５とし、撹乱を行わない場合のＸＰｔｅｘｔＬ（３７２４）に代入される値を、ＰｔｅｘｔＬ５、撹乱を行った場合の値をＸＰｔｅｘｔＬ５とすると、

ＸＰｔｅｘｔＲ５＝ＰｔｅｘｔＲ５　ｘｏｒ　ＰＸｏ２　（式７５）
ＸＰｔｅｘｔＬ５＝ＰｔｅｘｔＬ５　ｘｏｒ　ＰＸｏ１　（式７６）

となり、（式７５）、（式７６）は（式５１）、（式５２）と同様の変形が行われているため、次のラウンドの処理が図３４の実施例で行うことが出来る。

【００８６】

図３８は、図３７の処理とほぼ同様であるが、最後にＸＰｔｅｘｔＬとＸＰｔｅｘｔＲの交換を行っていない。したがって、図３８の撹乱を行わない場合のＸＰｔｅｘｔＲ（３８２２）に代入される値を、ＰｔｅｘｔＲ６、撹乱を行った場合の値をＸＰｔｅｘｔＲ６とし、撹乱を行わない場合のＸＰｔｅｘｔＬ（３８２４）に代入される値を、ＰｔｅｘｔＬ６、撹乱を行った場合の値をＸＰｔｅｘｔＬ６とすると、

ＸＰｔｅｘｔＲ６＝ＰｔｅｘｔＲ６　ｘｏｒ　ＰＸｏ１　（式７７）
ＸＰｔｅｘｔＬ６＝ＰｔｅｘｔＬ６　ｘｏｒ　ＰＸｏ２　（式７８）

となる。

【００８７】

図３９は、最終結果を求めるための処理フローを示す。第５逆変換処理（３９０５）により、転置済み平文撹乱用データ１のＰＸｏ２（３９０４）を用いて、転置済み変形平文１のＸＰｔｅｘｔＬ（３９０１）を逆変形し、第６逆変換処理（３９０６）により、転置済み平文撹乱用データ２のＰＸｏ１（３９０３）を用いて、転置済み変形平文２のＸＰｔｅｘｔＲ（３９０２）を逆変形する。これにより、変形がすべて取り除かれる。

ＰｔｅｘｔＲ６＝ＸＰｔｅｘｔＲ６　ｘｏｒ　ＰＸｏ１　（式７９）

ＰｔｅｘｔＬ６＝ＸＰｔｅｘｔＬ６　ｘｏｒ　ＰＸｏ２　（式８０）

最後に、転置処理ＩＰ－１（３９０７）を用いて、第５逆変換処理（３９０５）の結果と第６逆変換処理（３９０６）の結果を入力として、転置処理を行い、最終的な暗号文データＣｔｅｘｔ（３９０８）を得る。最終的な暗号データＣｔｅｘｔを得る直前まで、どの時点をとっても、データは変形を受けたままとなるので、電流波形から本来のデータを推測することが困難に成る。

【００８８】

ＳＢＯＸ表アドレス撹乱用データＳｉｎＸ、ＳＢＯＸ表データ撹乱用データＳｏｕｔＸおよび変形ＳＢＯＸ表の作成を、図１９もしくは図２０もしくは図２１もしくは図２２のような実施例で行ない、ハミングウエイトが常に一定とし、電流波形から本来のデータを推測することがさらに困難となった実施例が、請求項２８の実施例となる。

【００８９】

請求項２９の一実施例を、図３９、４０、４１、４２、４３、４４、４５、４６、４７、５２に示す。基本的な手順は請求項２７の実施例と同じであるが、ＳＢＯＸの変形を２回行っている点が、実施例２７と違う。ＳＢＯＸの変形の一実施例を図５２に示す。撹乱用データとしては、ＳＢＯＸアドレス撹乱用データＳｉｎＸ１（５２０１）、転置済みＳＢＯＸデータ撹乱用データＸＳｏｕｔ１（５２１０）、ＳＢＯＸアドレス撹乱用データＳｉｎＸ２（５２１２）、転置済みＳＢＯＸデータ撹乱用データＸＳｏｕｔＸ２（５２１８）を用い、第２変形済みＳＢＯＸ表（５２１４）を生成する。ＳＢＯＸ撹乱用データの個数が増えたため、図４０に示される、ラウンド１、ラウンド５、ラウンド９、ラウンド１３の処理の一実施例において、ＳＢＯＸアドレス撹乱用データＳｉｎＸ２（４０１５）を用いて、第３変形処理（４０１４）を行う処理が追加となり、図４１に示される、ラウンド２、ラウンド６、ラウンド１０、ラウンド１４の処理の一実施例において、ＳＢＯＸアドレス撹乱用データＳｉｎＸ２（４１１５）を用いて、第３変形処理（４１１４）を行う処理と、転置済みＳＢＯＸデータ撹乱用データＸＳｏ

ｕｔＸ２（４１２１）を用いて、第４逆変形処理（４１２０）が追加となり、図４２に示される、ラウンド３、ラウンド７、ラウンド１１、ラウンド１５の処理の一実施例において、ＳＢＯＸアドレス撹乱用データＳｉｎＸ２（４２１５）を用いて、第３変形処理（４２１４）を行う処理と、転置済みＳＢＯＸデータ撹乱用データＸＳｏｕｔＸ２（４２２１）を用いて、第４逆変形処理（４２２０）が追加となり、図４３に示される、ラウンド４、ラウンド８、ラウンド１２の処理の一実施例において、ＳＢＯＸアドレス撹乱用データＳｉｎＸ２（４３１５）を用いて、第３変形処理（４３１４）を行う処理が追加となり、図４４に示される、ラウンド１６の処理の一実施例において、ＳＢＯＸアドレス撹乱用データＳｉｎＸ２（４４１５）を用いて、第３変形処理（４４１４）を行う処理が追加となっている。

【００９０】

ＳＢＯＸ表アドレス撹乱用データＳｉｎＸ１、ＳＢＯＸ表アドレス撹乱用データＳｉｎＸ２、ＳＢＯＸ表データ撹乱用データＳｏｕｔＸ１、ＳＢＯＸ表データ撹乱用データＳｏｕｔＸ２及び第２変形ＳＢＯＸ表の作成を、図２６もしくは図２７もしくは図２８のような実施例で行った場合が、請求項３０の実施例となる。

【００９１】

ＳＢＯＸ表アドレス撹乱用データＳｉｎＸ１、ＳＢＯＸ表アドレス撹乱用データＳｉｎＸ２、ＳＢＯＸ表データ撹乱用データＳｏｕｔＸ１、ＳＢＯＸ表データ撹乱用データＳｏｕｔＸ２及び第２変形ＳＢＯＸ表の作成を、図２６もしくは図２７もしくは図２８のような実施例で行ない、かつハミングウエイトの検査を、全ビット通じてのハミングウエイトではなく、中央演算処理装置で一度に処理できるビット数に区切って評価した際にも、ハミングウエイトのが一定となるようにハミングウエイトの検査を実施したものが、請求項３１の実施例となる。

【００９２】

【発明の効果】

本頑発明によれば、ＩＣカードチップでの処理データを変形することと、その変形に用いる撹乱用データの生成時に制約を加えることにより、消費電流の波形

から、処理や暗号鍵の推測を困難にする。

【図面の簡単な説明】

【図１】

図１は、ＩＣカードのハードウエア構成の例を示す図である。

【図２】

図２は、ＩＣカード用チップ内のハードウエア構成の例を示す図である。

【図３】

図３は、消費電流の波形例を示す図である。

【図４】

図４は、一つの撹乱用データを使ったデータ変形の手順の例を示すフローチャートである。

【図５】

図５は、撹乱用データをあらかじめ複数用意し、選択することで撹乱用データを生成する、一つの撹乱用データを使ったデータ変形の手順の例を示すフローチャートである。

【図６】

図６は、ハミングウエイト一定の乱数生成方法を示すフローチャートである。

【図７】

図７は、ハミングウエイト一定の乱数生成方法の例を示すフローチャートである。

【図８】

図８は、ハミングウエイト一定の値の一覧表作成方法の例を示すフローチャートである。

【図９】

図９は、ハミングウエイトが一定でかつ、データ処理手順処理後のハミングウエイト一定の値の一覧表作成方法の例を示すフローチャートである。

【図１０】

図１０は、ビット長の短いハミングウエイト一定の一覧表から、ビット長の長いハミングウエイト一定の乱数の生成方法の例を示すフローチャートである。

【図１１】

図１１は、撹乱用データおよび処理済撹乱用データ生成方法の例を示すフローチャートである。

【図１２】

図１２は、撹乱用データおよび処理済撹乱用データ生成方法の例を示すフローチャートである。

【図１３】

図１３は、表引きによるデータ処理を、２つの撹乱用データを用いてタ変形して処理する方法の例を示すフローチャートである。

【図１４】

図１４は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図１５】

図１５は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図１６】

図１６は、撹乱用データおよび変形済み表生成方法。

【図１７】

表引きと１つのデータ処理手段からなるデータ処理を、２つの撹乱用データを用いて変形して処理する方法の例を示すフローチャートである。

【図１８】

図１８は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図１９】

図１９は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２０】

図２０は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２１】

図２１は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２２】

図２２は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２３】

図２３は、表引きと１つのデータ処理手段からなる処理を、２回繰り返すデータ処理を、２つの撹乱用データを用いて変形して処理する方法の例を示すフローチャートである。

【図２４】

図２４は、表引きと１つのデータ処理手段からなる処理を、２回繰り返すデータ処理を、４つの撹乱用データを用いて変形して処理する方法の例を示すフローチャートである。

【図２５】

図２５は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２６】

図２６は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２７】

図２７は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２８】

図２８は、撹乱用データおよび変形済み表生成方法の例を示すフローチャートである。

【図２９】

図２９は、ＤＥＳ処理用ＳＢＯＸ撹乱用データおよび変形ＳＢＯＸ生成方法の例を示すフローチャートである。

【図３０】

図３０は、ＤＥＳ処理用平文撹乱用データ生成方法の例を示すフローチャートである。

【図３１】

図３１は、ＤＥＳ処理用秘密鍵撹乱用データ生成方法の例を示すフローチャートである。

【図３２】

図３２は、平文変形方法の例を示すフローチャートである。

【図３３】

図３３は、秘密鍵変形方法の例を示すフローチャートである。

【図３４】

図３４は、ＤＥＳ第１、第５、第９、第１３ラウンド処理方法の例を示すフローチャートである。

【図３５】

図３５は、ＤＥＳ第２、第６、第１０、第１４ラウンド処理方法の例を示すフローチャートである。

【図３６】

図３６は、ＤＥＳ第３、第７、第１１、第１５ラウンド処理方法の例を示すフローチャートである。

【図３７】

図３７は、ＤＥＳ第４、第８、第１２ラウンド処理方法の例を示すフローチャートである。

【図３８】

図３８は、ＤＥＳ第１６ラウンド処理方法の例を示すフローチャートである。

【図３９】

図３９は、ＤＥＳ最終逆変形方法の例を示すフローチャートである。

【図４０】

図４０は、ＤＥＳ第１、第５、第９、第１３ラウンド処理方法の例を示すフローチャートである。

【図４１】

図４１は、ＤＥＳ第２、第６、第１０、第１４ラウンド処理方法の例を示すフローチャートである。

【図４２】

図４２は、ＤＥＳ第３、第７、第１１、第１５ラウンド処理方法の例を示すフローチャートである。

【図４３】

図４３は、ＤＥＳ第４、第８、第１２ラウンド処理方法の例を示すフローチャートである。

【図４４】

図４４は、ＤＥＳ第１６ラウンド処理方法の例を示すフローチャートである。

【図４５】

図４５は、ＳＢＯＸアクセス方法の例を示すフローチャートである。

【図４６】

図４６は、ＳＢＯＸ表の例を示す図である。

【図４７】

図４６は、ＳＢＯＸ表変形方法の例を示すフローチャートである。

【図４８】

図４８は、ハミングウエイト一定撹乱用かつ処理済のハミングウエイトも一定のデータ生成方法の例を示すフローチャートである。

【図４９】

図４９は、ハミングウエイト一定乱数生成方式の例を示すフローチャートである。

【図５０】

図５０は、ハミングウエイト一定乱数生成方式の例を示すフローチャートである。

【図５１】

図５１は、ハミングウエイト一定乱数生成方式の例を示すフローチャートである。

【図５２】

図５２は、ＤＥＳ処理用ＳＢＯＸ攪乱用データおよび変形ＳＢＯＸ生成方法の例を示すフローチャートである。

【図５３】

図５３は、ＤＥＳ処理用攪乱用データ生成方法の例を示すフローチャートである。

【図５４】

図５４は、ＤＥＳ処理中間データ攪乱方法の例を示すフローチャートである

【図５５】

図５５は、図１２に対応するテーブルの例を示す図である。

【図５６】

図５６は、第１攪乱用データ格納手段（１５０１）に格納された第１攪乱用データ、及び第２攪乱用データ格納手段（１５０２）に格納された第２攪乱用データの例を示す図である。

【図５７】

図５７は、表格納手段（１５０７）に格納されたデータの例を示す図である。

【図５８】

図５８は、第１攪乱用データ、第２攪乱用データ及び変形済み表のテーブルの例を示す図である。

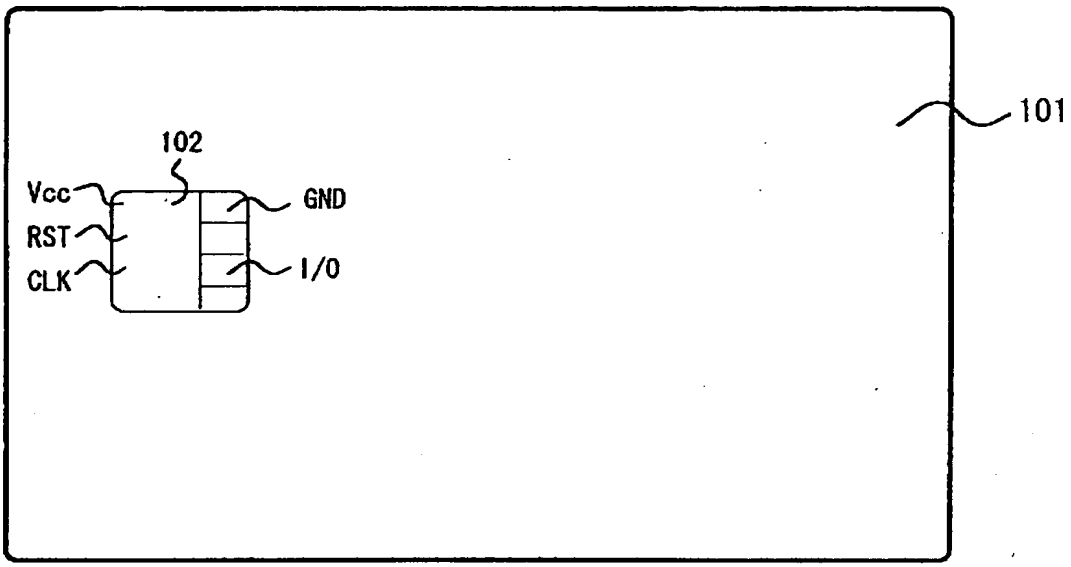
【符号の説明】

４０１：入力データ、４０２：データ変形処理手段、４０３：攪乱用データ、４０４：変形データ、４０５：変形データ処理手段、４０６：変形データ、４０７：変形データ、４０８：変形済み攪乱用データ、４０９：処理済データ、５０２：ハミングウエート一定の攪乱用データ、５０３：攪乱用データ処理手段、５０４：処理済攪乱用データ、５０５：ハミングウエイト検査手段、６０１：ｎビット乱数発生手段、６０２：ｎビット乱数、６０３：ビット反転処理手段、６０４：反転ｎビット乱数、６０５：データ統合手段、６０６：ハミングウエイト一定２ｎビット乱数。

【書類名】図面

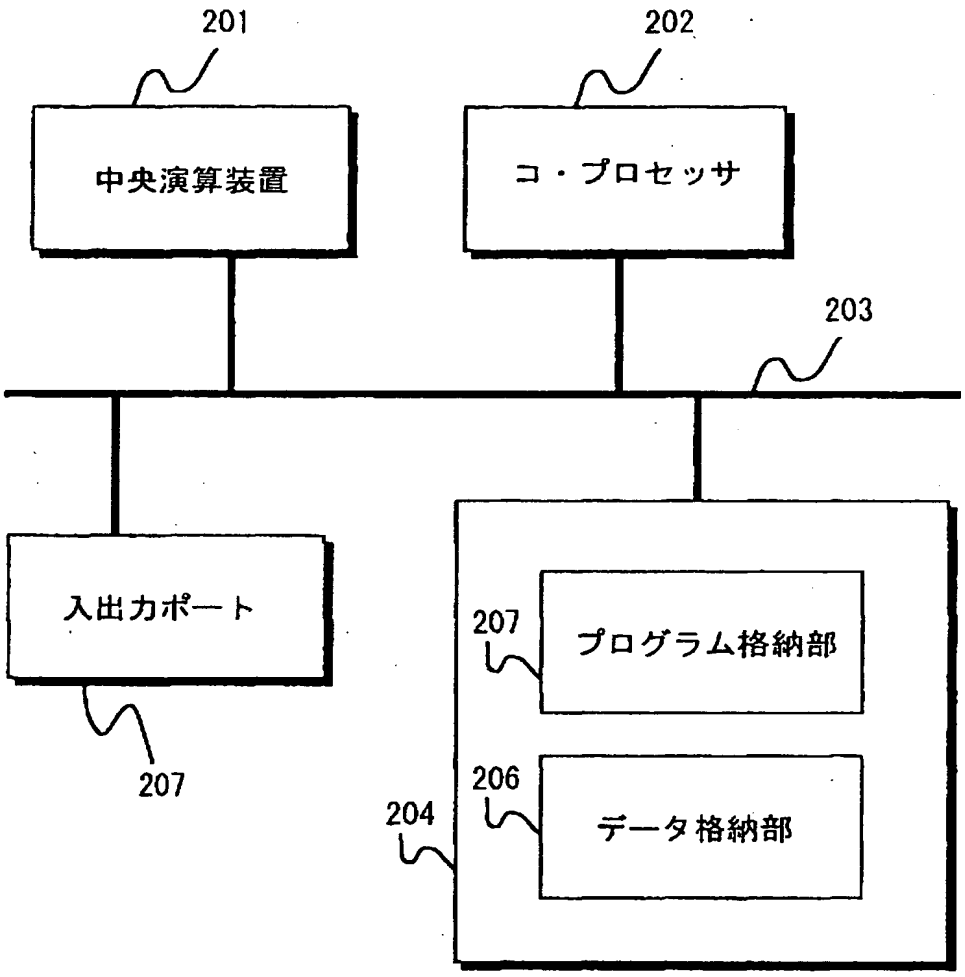
【図１】

図　1



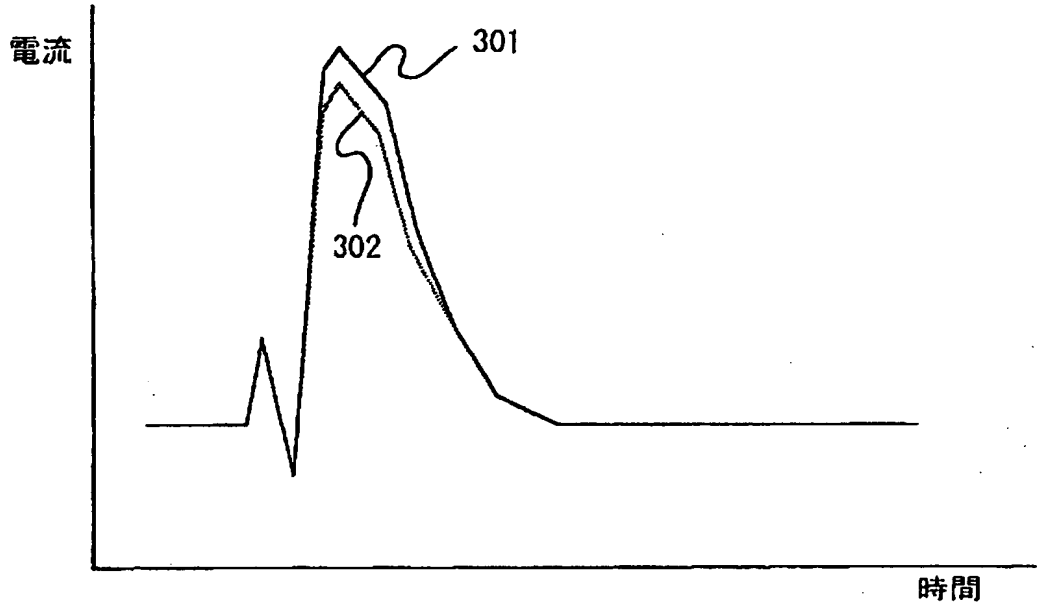
【図２】

図　２


201
202
中央演算装置
コ・プロセッサ
203
入出力ポート
207
207
プログラム格納部
206
データ格納部
204

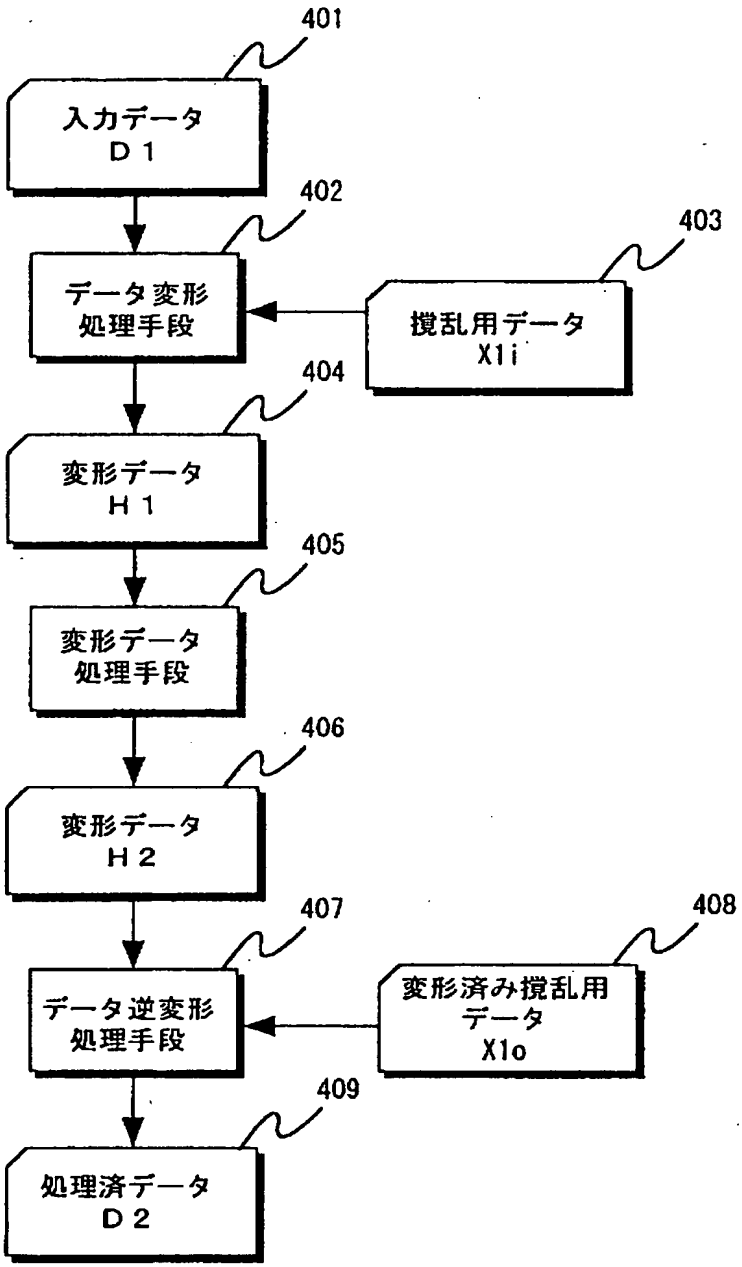
【図３】

図　3



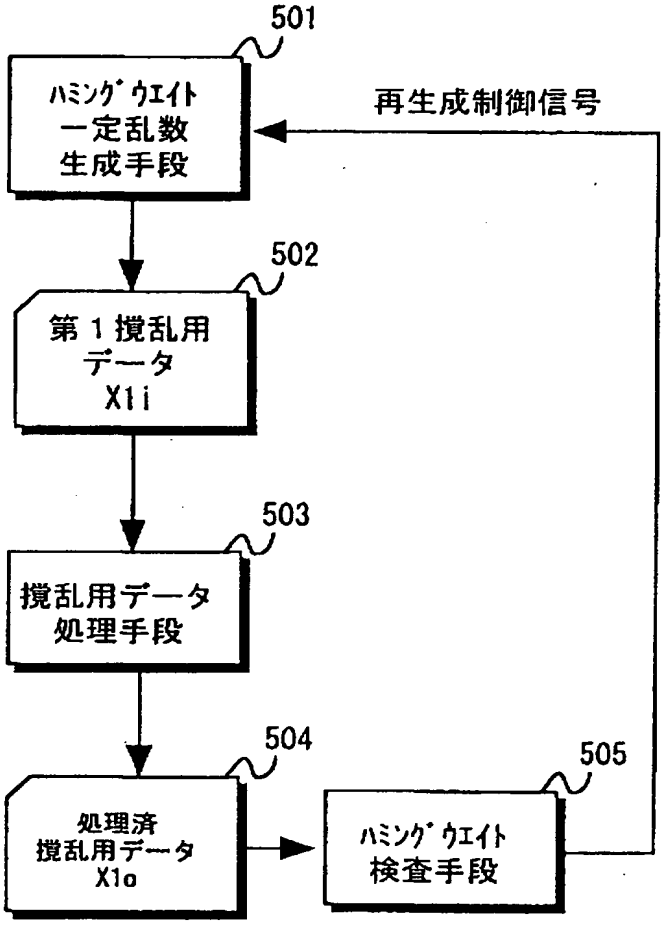
【図４】

図　4


401
入力データ
Ｄ１
402
データ変形
処理手段
403
撹乱用データ
X1i
404
変形データ
Ｈ１
405
変形データ
処理手段
406
変形データ
Ｈ２
407
データ逆変形
処理手段
408
変形済み撹乱用
データ
X1o
409
処理済データ
Ｄ２

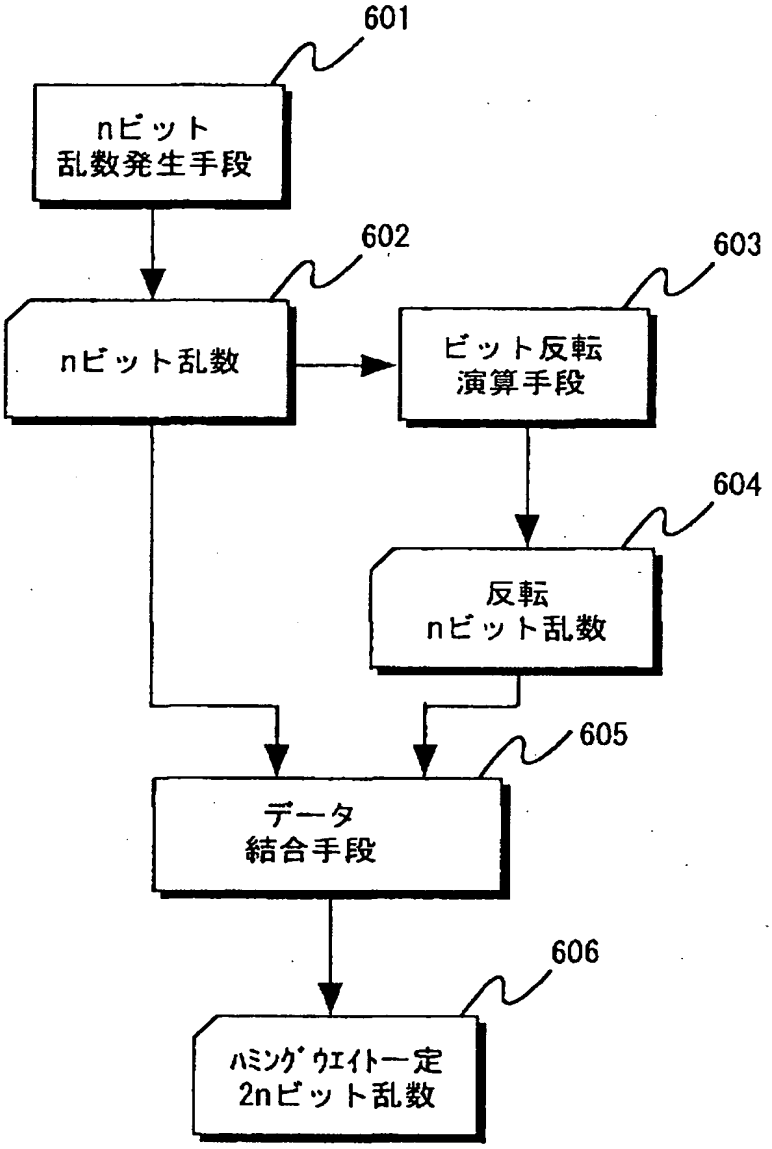
【図５】

図　5


501
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
再生成制御信号
502
第１撹乱用
データ
X1i
503
撹乱用データ
処理手段
504
処理済
撹乱用データ
X1o
505
ﾊﾐﾝｸﾞｳｴｲﾄ
検査手段

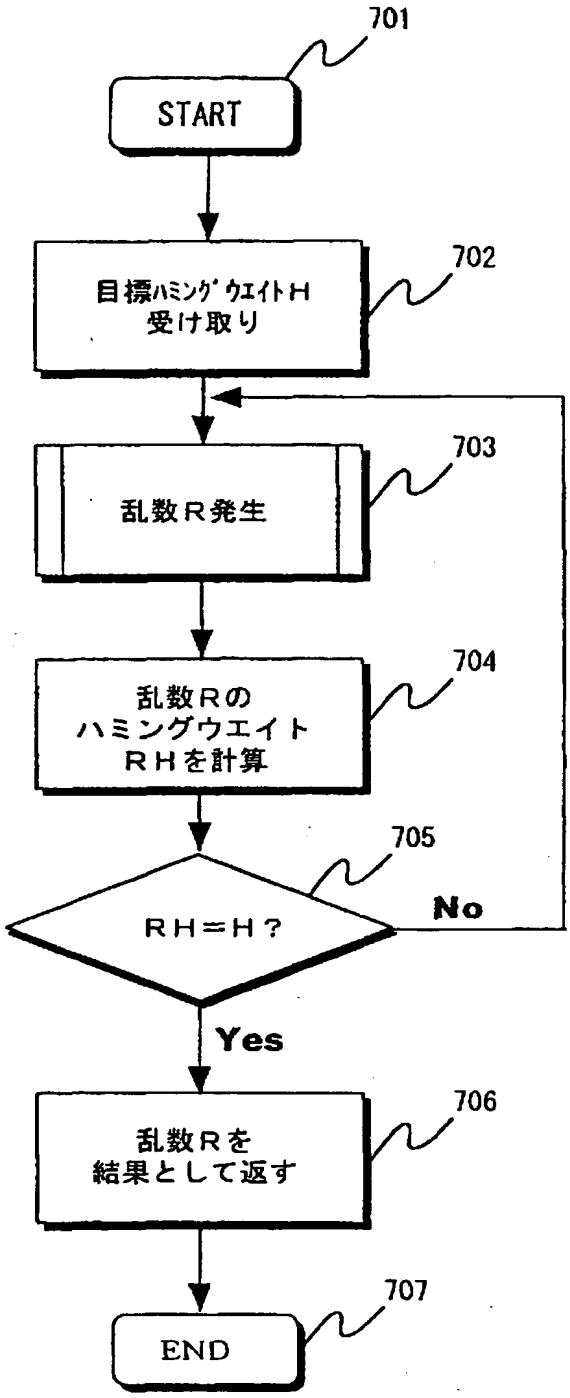
【図６】

図　6


601
nビット
乱数発生手段
602
nビット乱数
603
ビット反転
演算手段
604
反転
nビット乱数
605
データ
結合手段
606
ﾊﾐﾝｸﾞｳｴｲﾄ一定
2nビット乱数

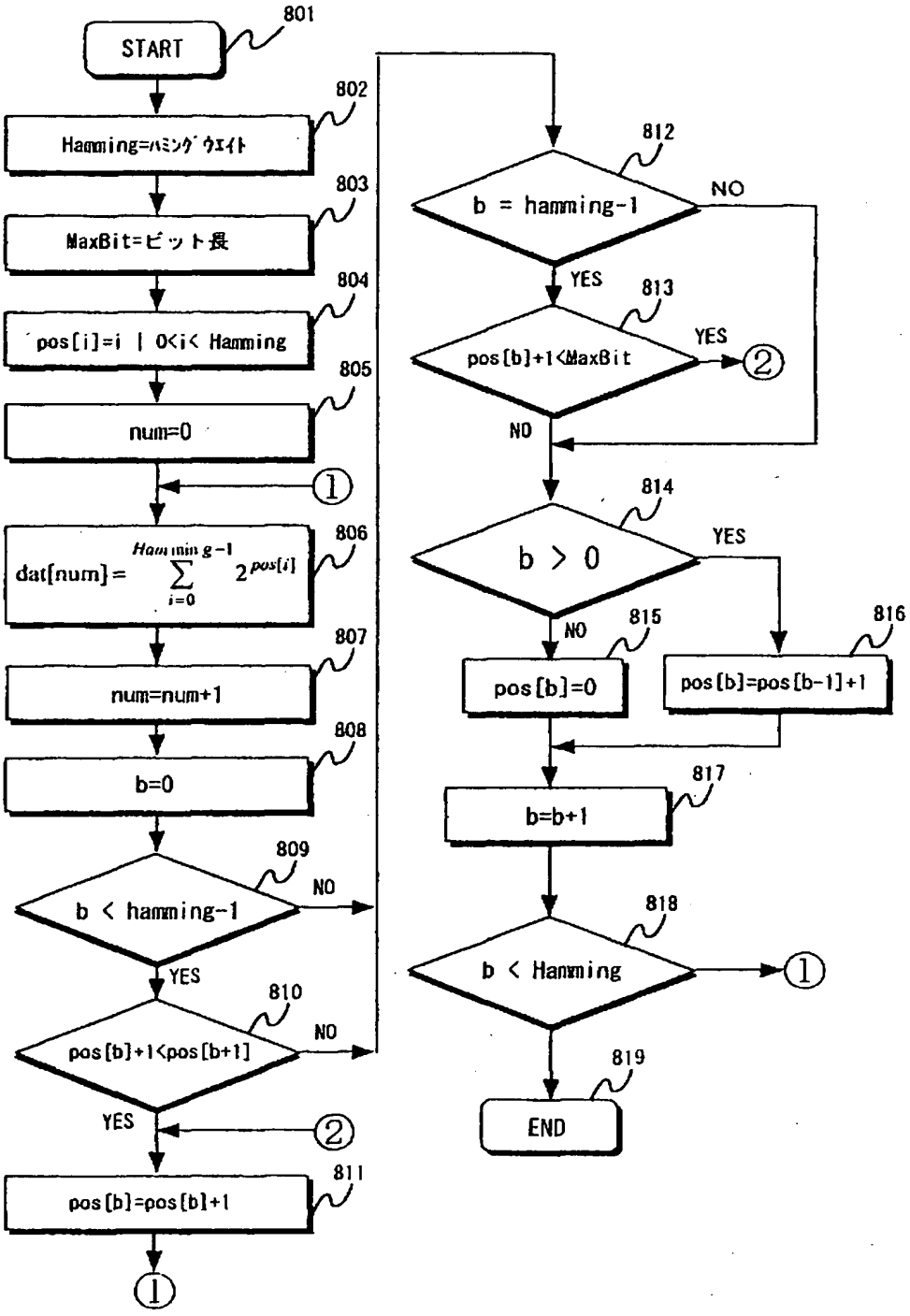
【図７】

図　7


701
START
目標ﾊﾐﾝｸﾞｳｴｲﾄH
受け取り
702
乱数Ｒ発生
703
乱数Ｒの
ハミングウエイト
ＲＨを計算
704
ＲＨ＝Ｈ？
705
No
Yes
乱数Ｒを
結果として返す
706
END
707

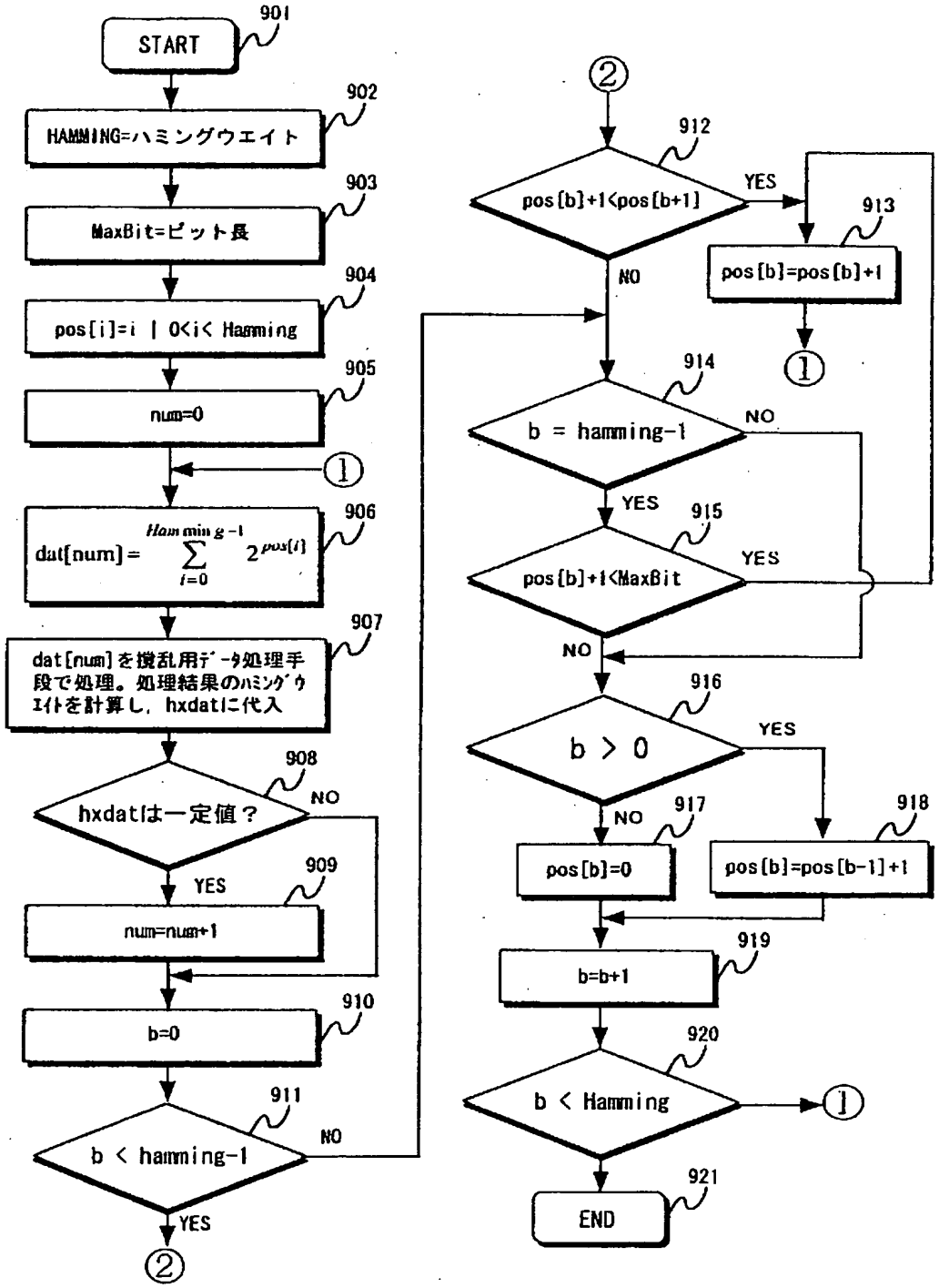
【図８】

図　8


801
START
802
Hamming=ハミング ウエイト
803
MaxBit=ビット長
804
pos[i]=i | 0<i< Hamming
805
num=0
①
806
dat[num]= Σ 2^pos[i]
807
num=num+1
808
b=0
809
b < hamming-1
NO
YES
810
pos[b]+1<pos[b+1]
NO
YES
②
811
pos[b]=pos[b]+1
①
812
b = hamming-1
NO
YES
813
pos[b]+1<MaxBit
YES
②
NO
814
b > 0
YES
NO
815
pos[b]=0
816
pos[b]=pos[b-1]+1
817
b=b+1
818
b < Hamming
①
819
END

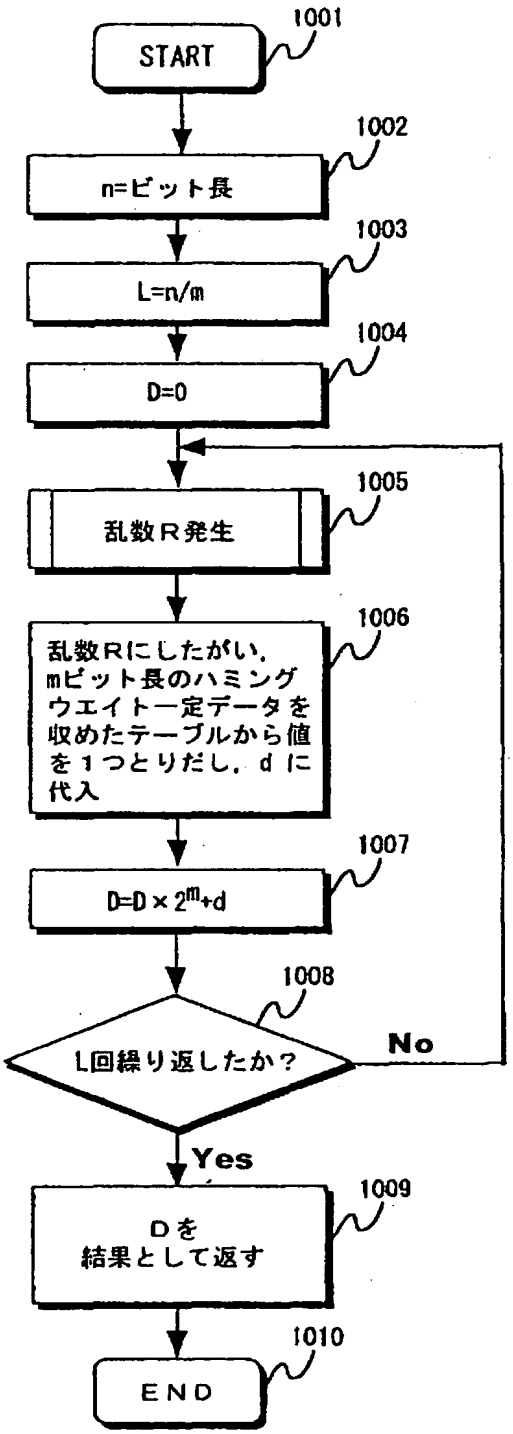
【図９】

図　9


START
901
HAMMING=ハミングウエイト
902
MaxBit=ビット長
903
pos[i]=i | 0<i< Hamming
904
num=0
905
①
dat[num]= Σ 2^pos[i]
906
dat[num]を攪乱用データ処理手段で処理。処理結果のハミングウエイトを計算し，hxdatに代入
907
hxdatは一定値？
908
NO
YES
num=num+1
909
b=0
910
b < hamming-1
911
NO
YES
②
②
pos[b]+1<pos[b+1]
912
YES
pos[b]=pos[b]+1
913
NO
①
b = hamming-1
914
NO
YES
pos[b]+1<MaxBit
915
YES
NO
b > 0
916
YES
NO
pos[b]=0
917
pos[b]=pos[b-1]+1
918
b=b+1
919
b < Hamming
920
①
END
921

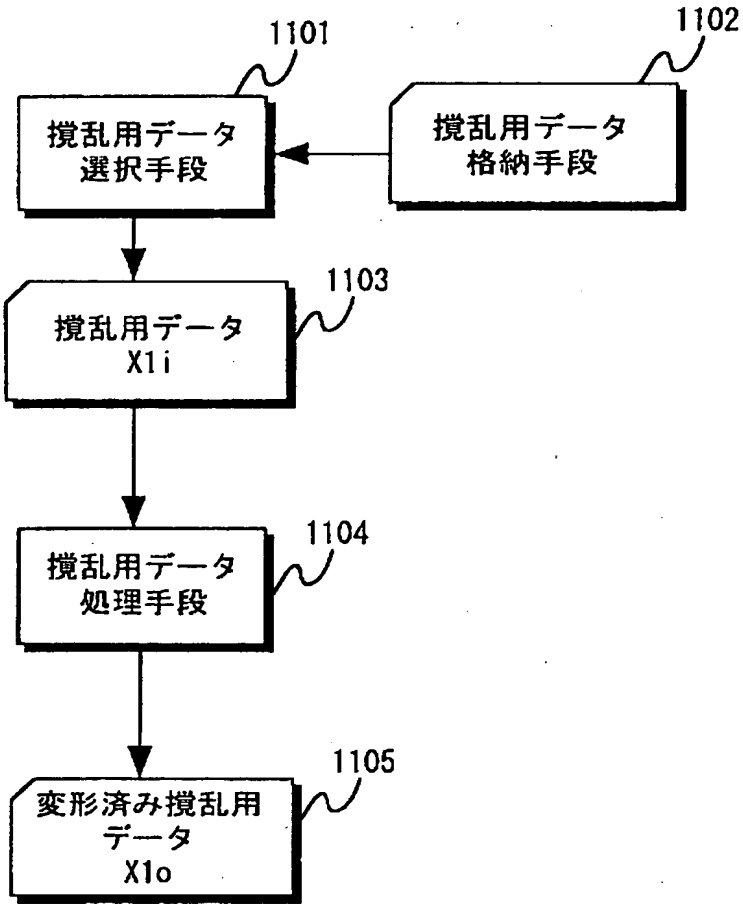
【図１０】

図　10


1001
START
1002
n=ビット長
1003
L=n/m
1004
D=0
1005
乱数Ｒ発生
1006
乱数Ｒにしたがい，mビット長のハミングウエイト一定データを収めたテーブルから値を１つとりだし，ｄに代入
1007
D=D×2^m+d
1008
L回繰り返したか？
No
Yes
1009
Ｄを結果として返す
1010
END

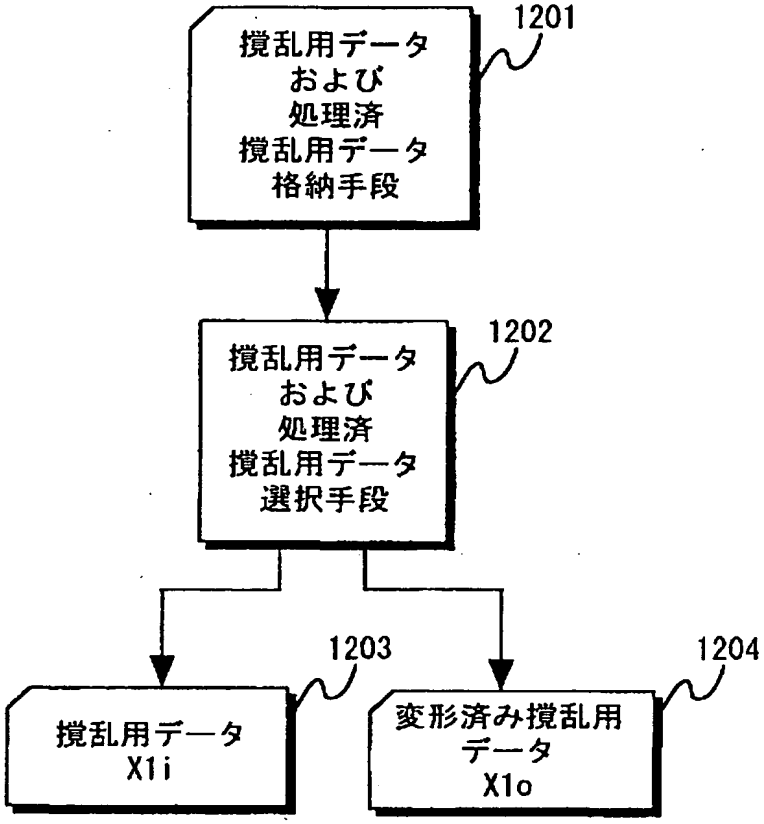
【図１１】

図　11


1101
1102
撹乱用データ
選択手段
撹乱用データ
格納手段
1103
撹乱用データ
X1i
1104
撹乱用データ
処理手段
1105
変形済み撹乱用
データ
X1o

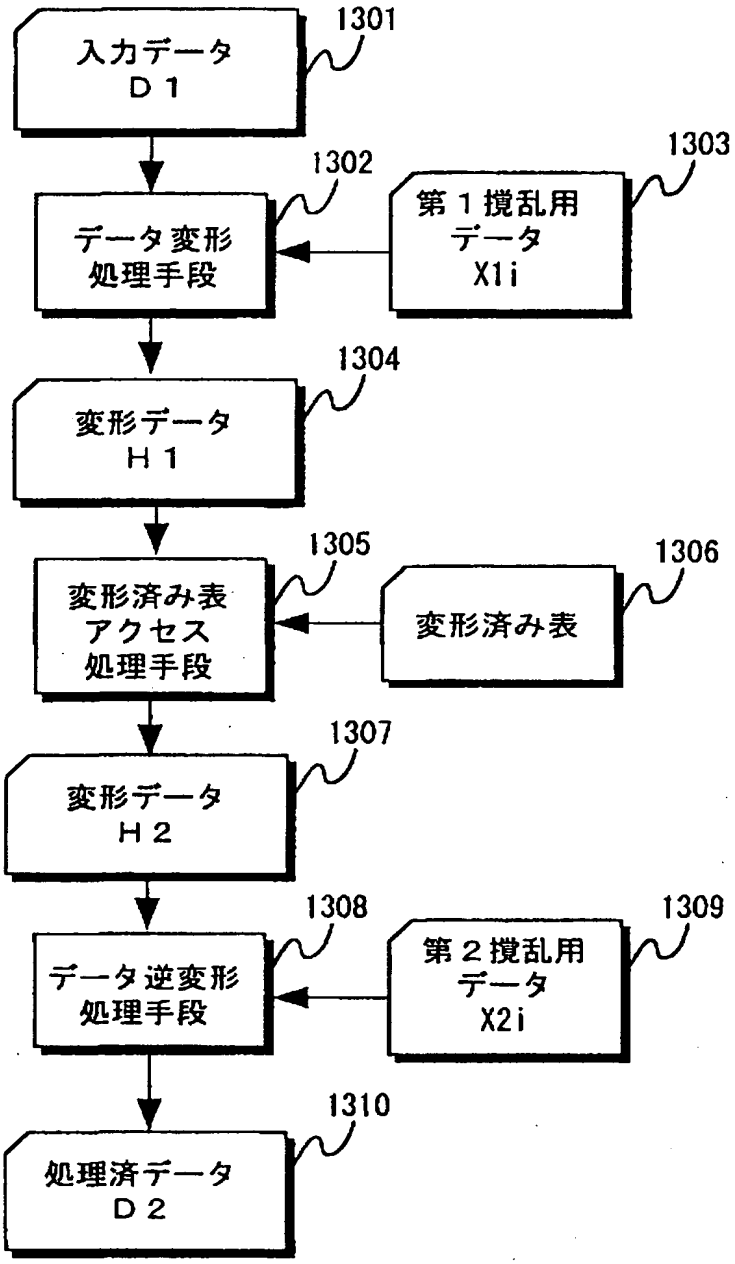
【図１２】

図　12


撹乱用データおよび処理済撹乱用データ格納手段
1201
撹乱用データおよび処理済撹乱用データ選択手段
1202
撹乱用データ X1i
1203
変形済み撹乱用データ X1o
1204

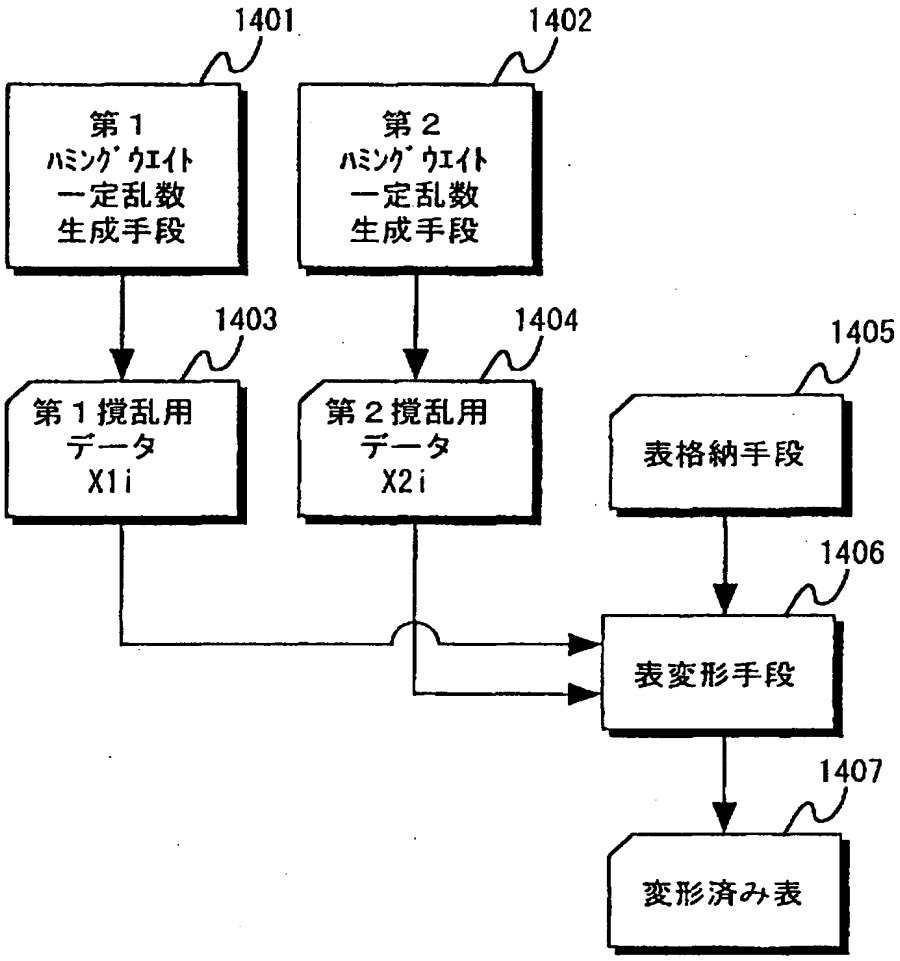
【図１３】

図　13


入力データ
Ｄ１
1301
データ変形
処理手段
1302
第１撹乱用
データ
X1i
1303
変形データ
Ｈ１
1304
変形済み表
アクセス
処理手段
1305
変形済み表
1306
変形データ
Ｈ２
1307
データ逆変形
処理手段
1308
第２撹乱用
データ
X2i
1309
処理済データ
Ｄ２
1310

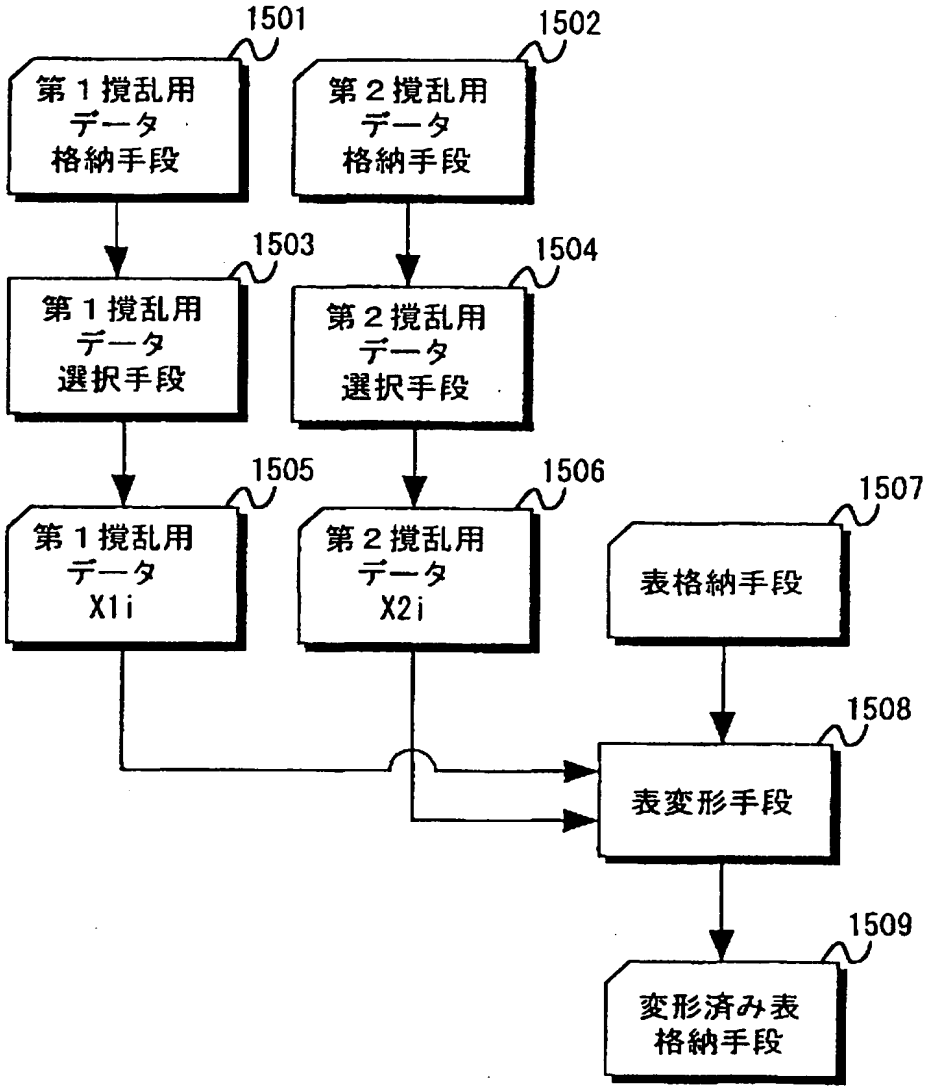
【図１４】

図　14


1401
1402
第１
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
第２
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
1403
1404
1405
第１撹乱用
データ
X1i
第２撹乱用
データ
X2i
表格納手段
1406
表変形手段
1407
変形済み表

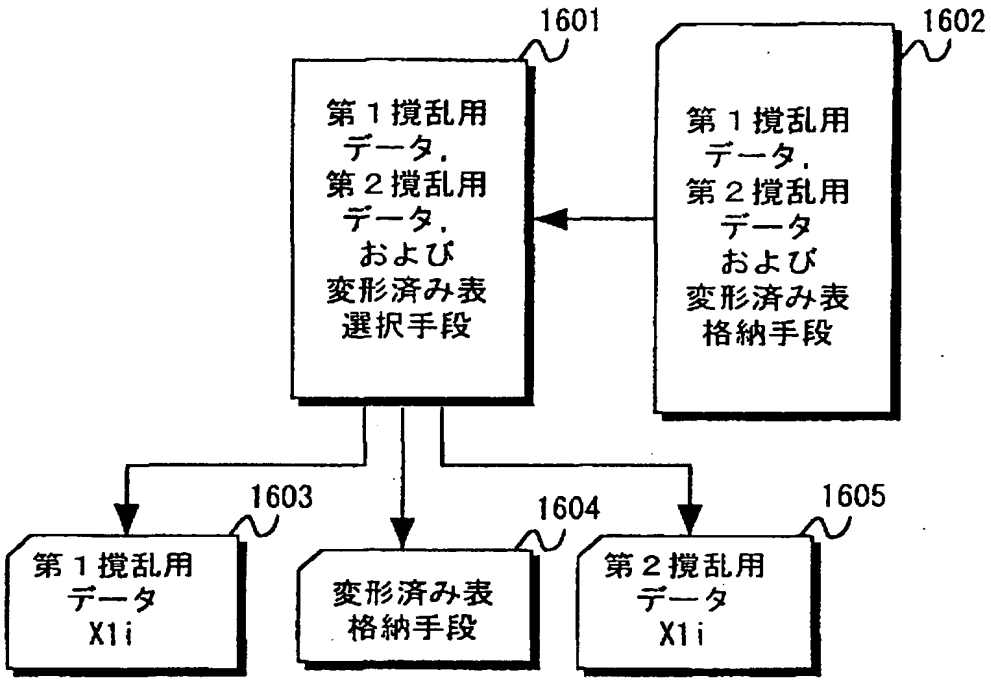
【図１５】

図　15


1501
第１撹乱用
データ
格納手段
1502
第２撹乱用
データ
格納手段
1503
第１撹乱用
データ
選択手段
1504
第２撹乱用
データ
選択手段
1505
第１撹乱用
データ
X1i
1506
第２撹乱用
データ
X2i
1507
表格納手段
1508
表変形手段
1509
変形済み表
格納手段

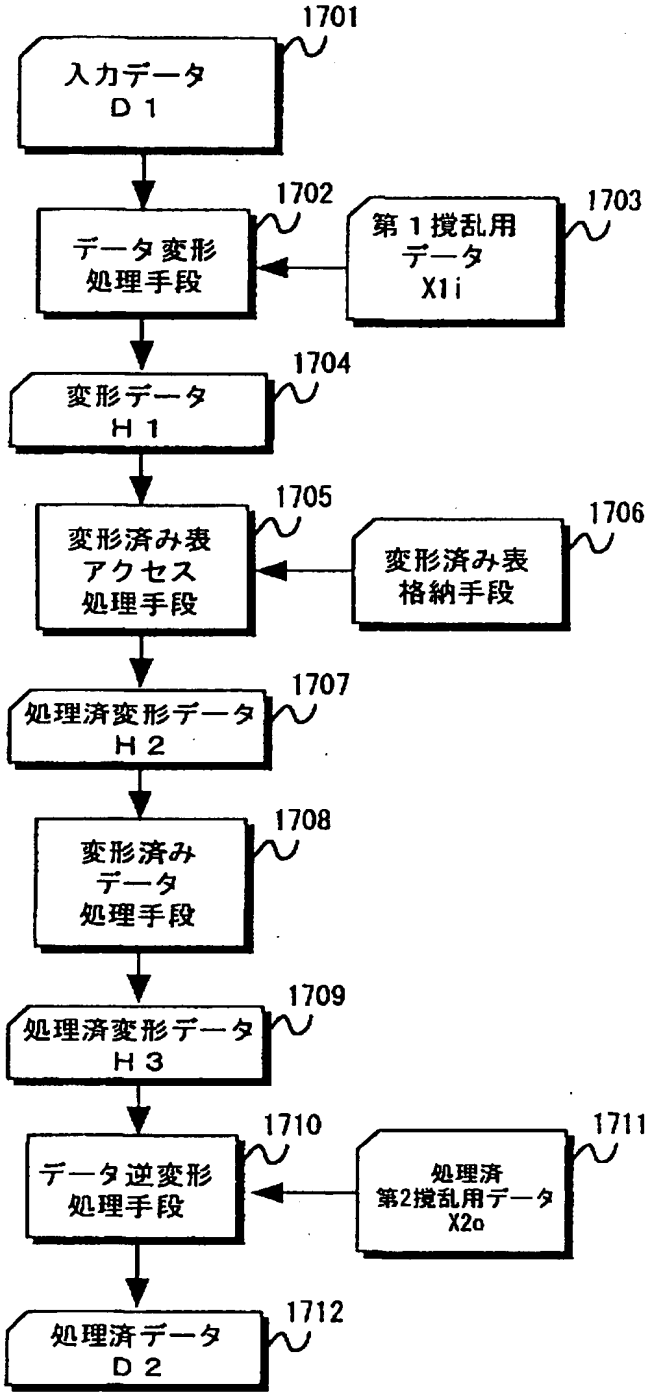
【図１６】

図　16


1601
第１撹乱用データ，第２撹乱用データ，および変形済み表選択手段
1602
第１撹乱用データ，第２撹乱用データおよび変形済み表格納手段
1603
第１撹乱用データX1i
1604
変形済み表格納手段
1605
第２撹乱用データX1i

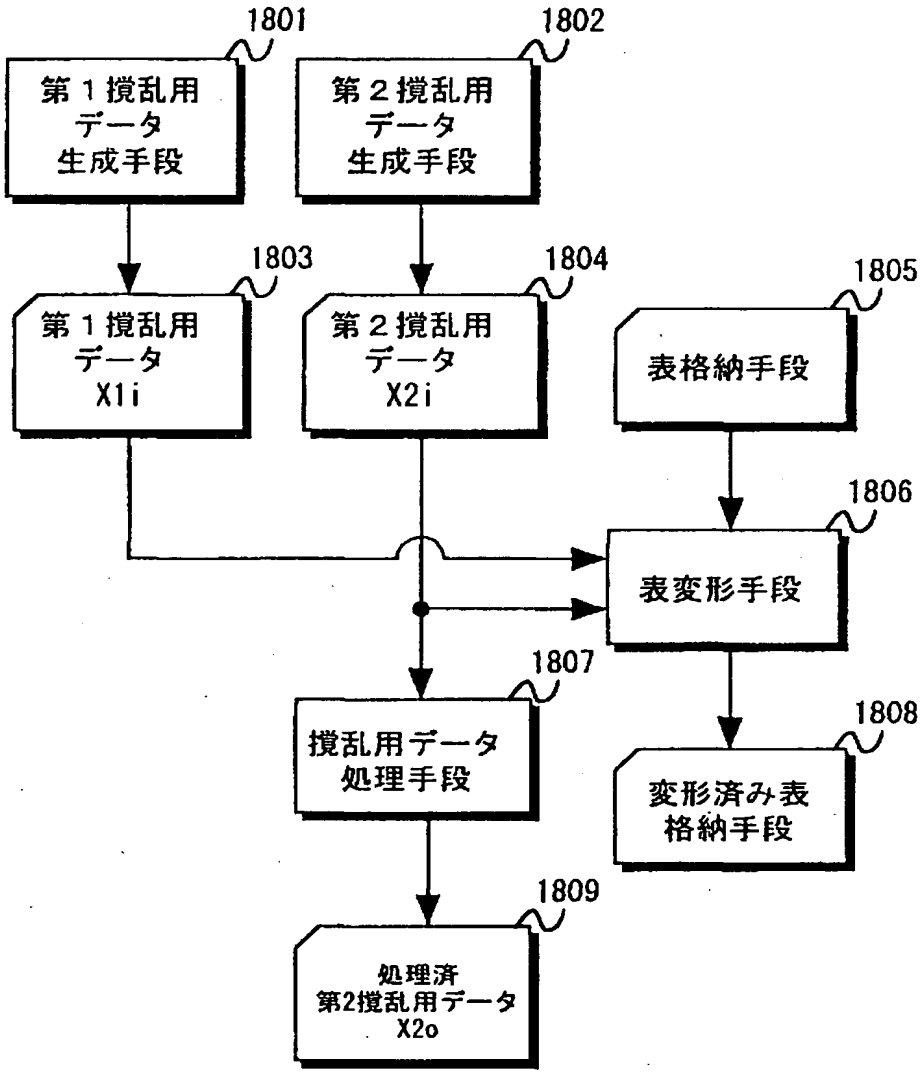
【図１７】

図　17


1701
入力データ
Ｄ１
1702
データ変形
処理手段
1703
第１撹乱用
データ
X1i
1704
変形データ
Ｈ１
1705
変形済み表
アクセス
処理手段
1706
変形済み表
格納手段
1707
処理済変形データ
Ｈ２
1708
変形済み
データ
処理手段
1709
処理済変形データ
Ｈ３
1710
データ逆変形
処理手段
1711
処理済
第2撹乱用データ
X2o
1712
処理済データ
Ｄ２

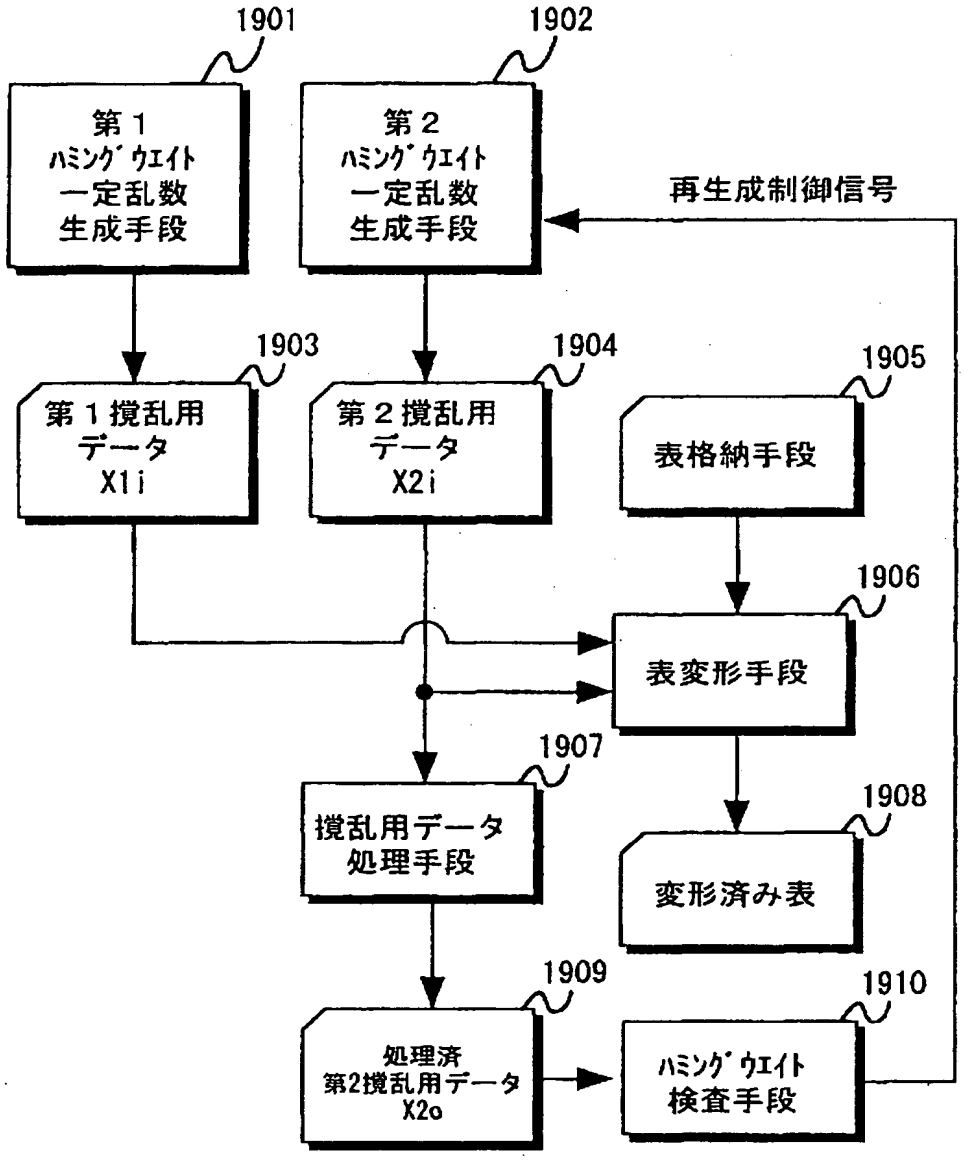
【図１８】

図　18


1801
第１撹乱用
データ
生成手段
1802
第２撹乱用
データ
生成手段
1803
第１撹乱用
データ
X1i
1804
第２撹乱用
データ
X2i
1805
表格納手段
1806
表変形手段
1807
撹乱用データ
処理手段
1808
変形済み表
格納手段
1809
処理済
第2撹乱用データ
X2o

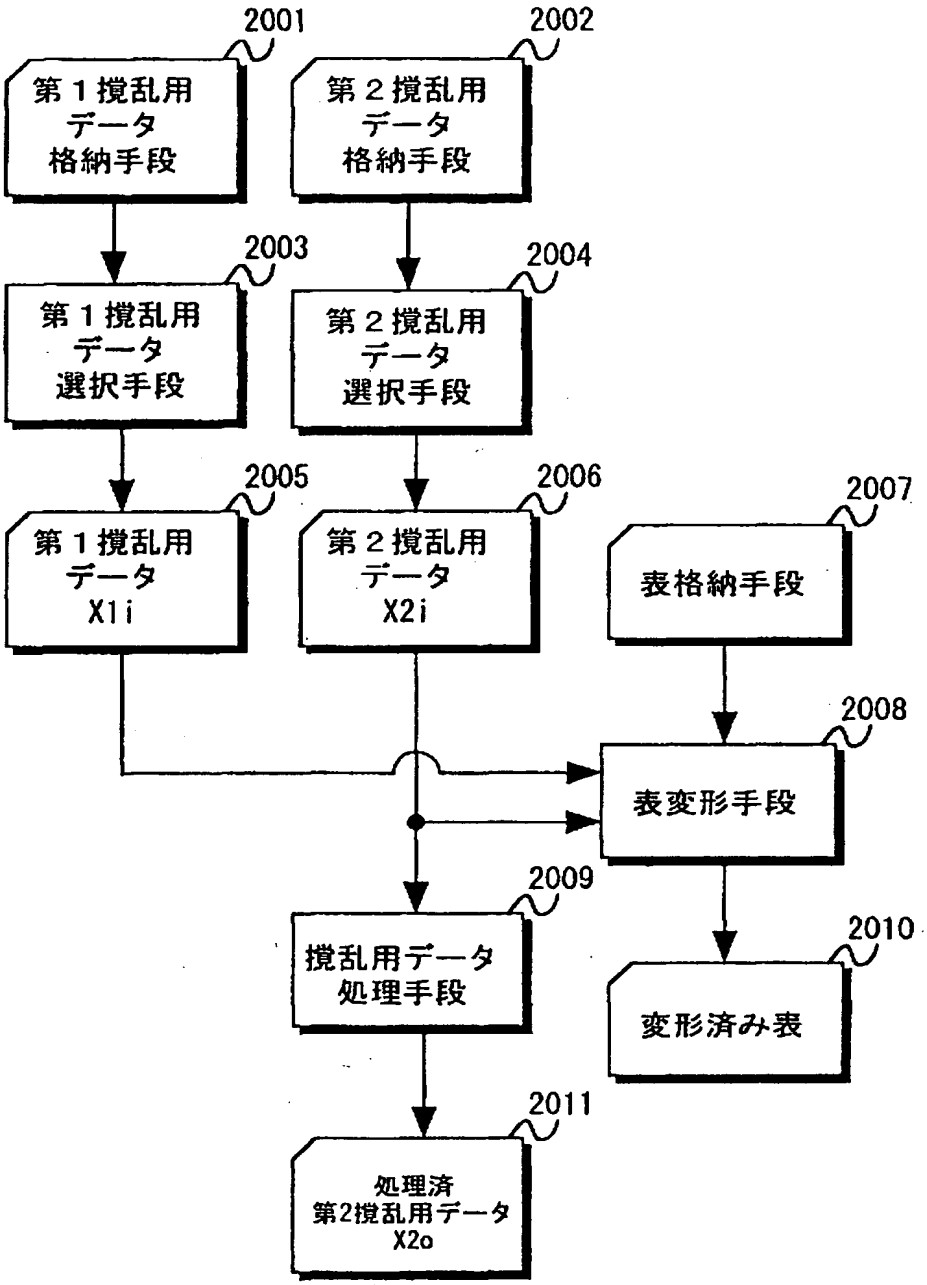
【図１９】

図 19


1901
1902
第１
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
第２
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
再生成制御信号
1903
1904
1905
第１撹乱用
データ
X1i
第２撹乱用
データ
X2i
表格納手段
1906
表変形手段
1907
1908
撹乱用データ
処理手段
変形済み表
1909
1910
処理済
第2撹乱用データ
X2o
ﾊﾐﾝｸﾞｳｴｲﾄ
検査手段

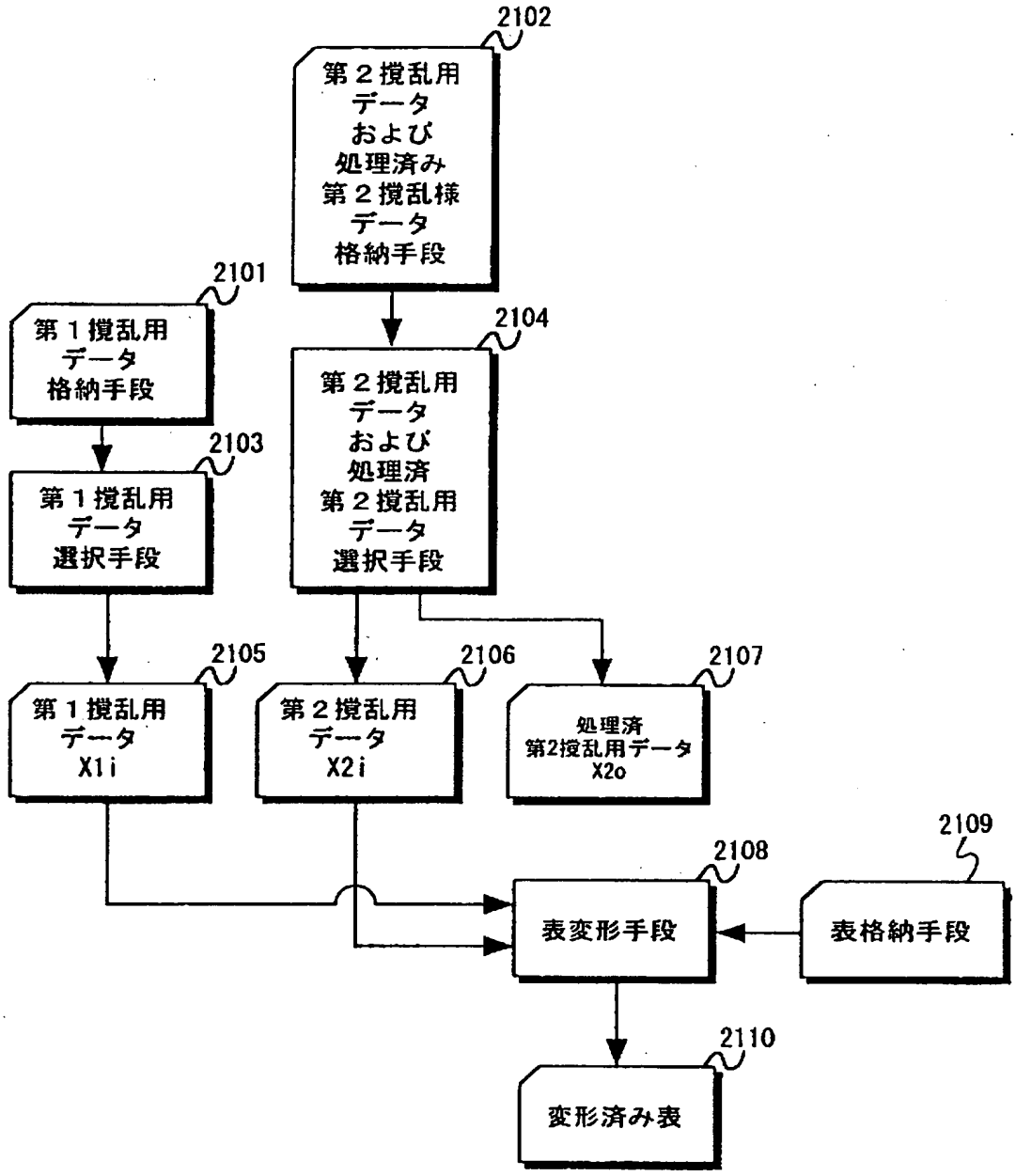
【図２０】

図 20


2001
第１撹乱用データ格納手段
2002
第２撹乱用データ格納手段
2003
第１撹乱用データ選択手段
2004
第２撹乱用データ選択手段
2005
第１撹乱用データ X1i
2006
第２撹乱用データ X2i
2007
表格納手段
2008
表変形手段
2009
撹乱用データ処理手段
2010
変形済み表
2011
処理済第2撹乱用データ X2o

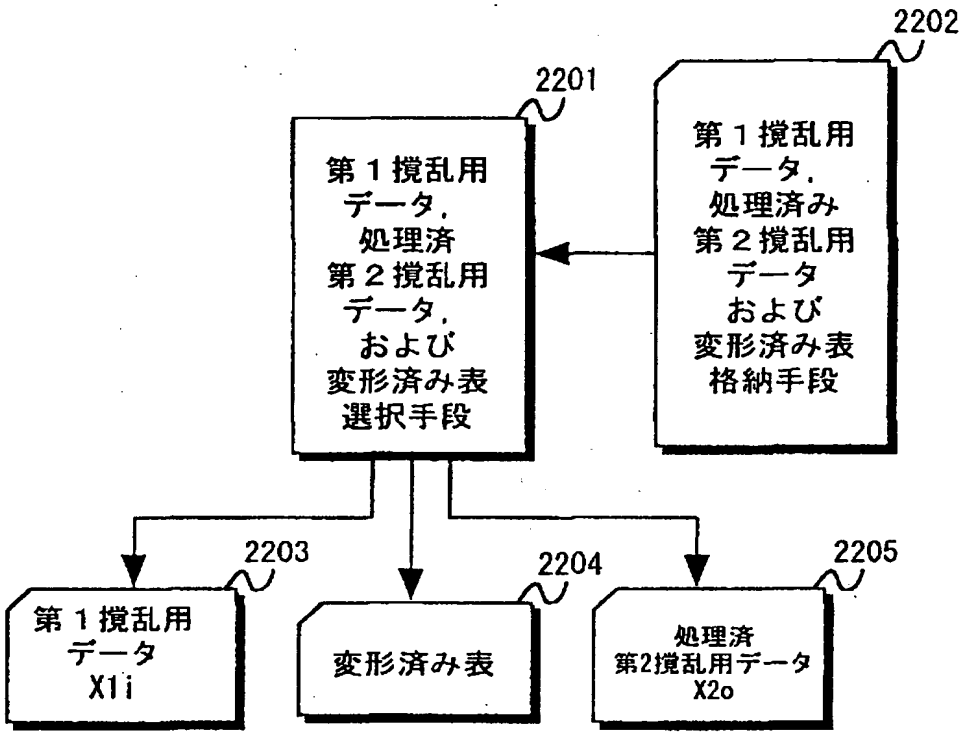
【図２１】

図　21


2102
第２撹乱用データおよび処理済み第２撹乱様データ格納手段
2101
第１撹乱用データ格納手段
2104
第２撹乱用データおよび処理済第２撹乱用データ選択手段
2103
第１撹乱用データ選択手段
2105
第１撹乱用データ X1i
2106
第２撹乱用データ X2i
2107
処理済第2撹乱用データ X2o
2109
2108
表変形手段
表格納手段
2110
変形済み表

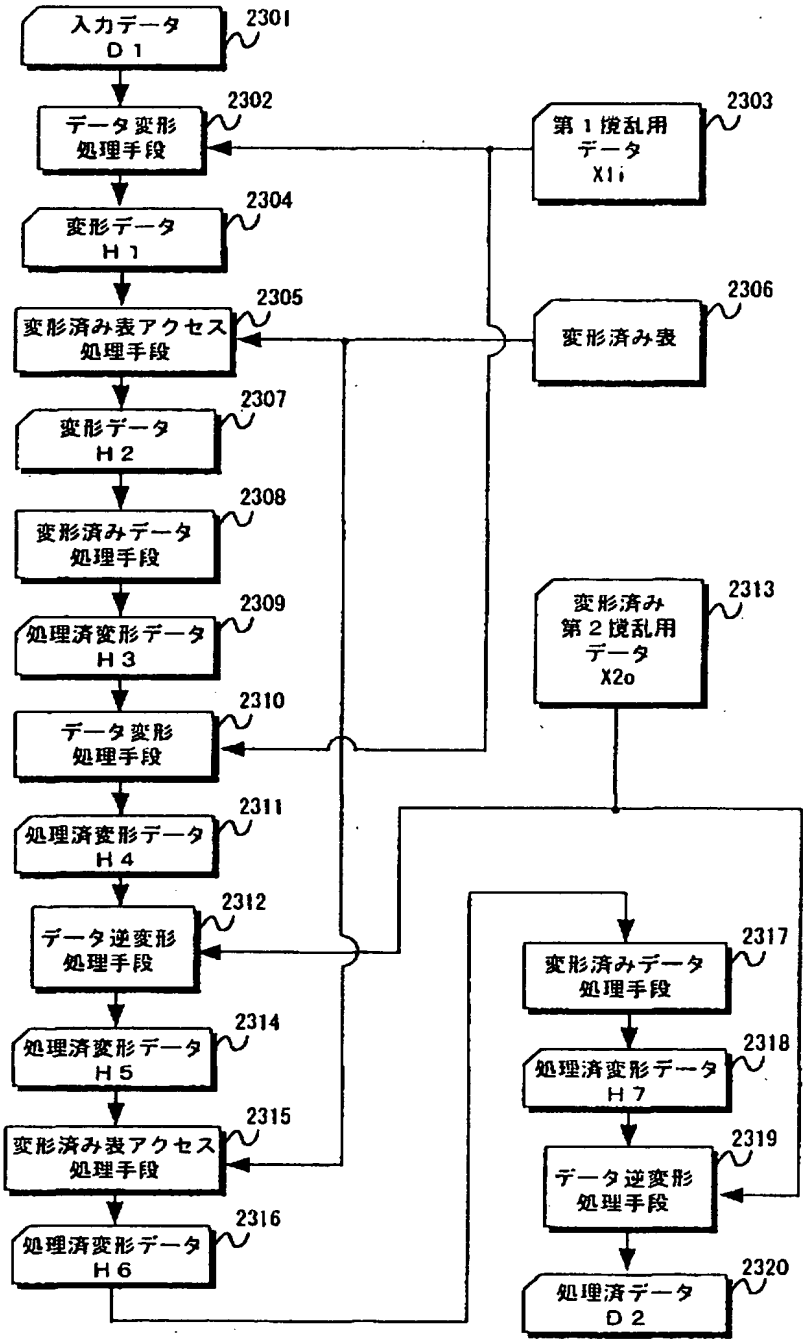
【図２２】

図　22


2202
2201
第１撹乱用データ，処理済第２撹乱用データ，および変形済み表選択手段
第１撹乱用データ，処理済み第２撹乱用データおよび変形済み表格納手段
2203
2204
2205
第１撹乱用データ X1i
変形済み表
処理済第2撹乱用データ X2o

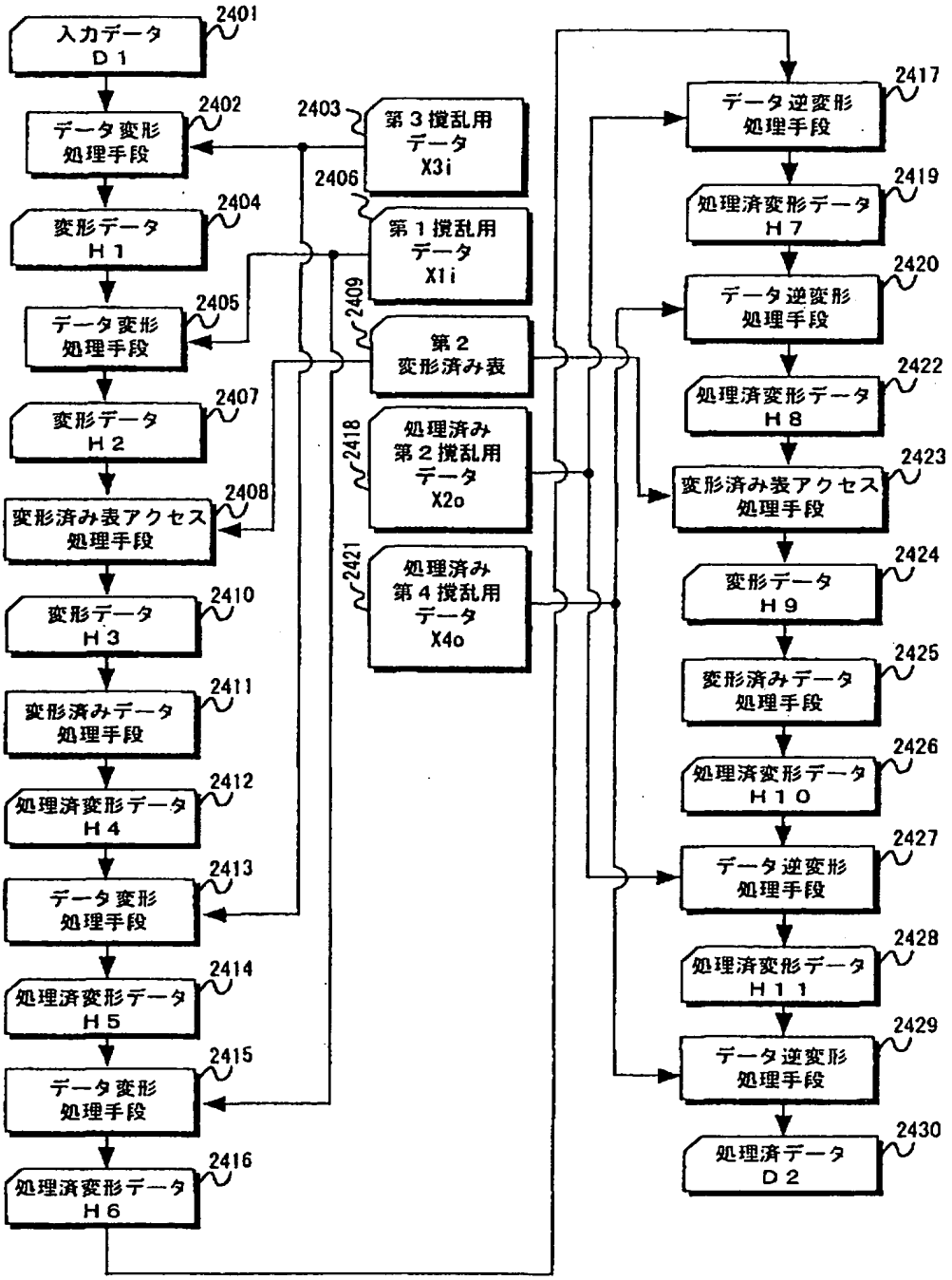
【図２３】

図 23


入力データ D1
2301
データ変形処理手段
2302
第1撹乱用データ X1i
2303
変形データ H1
2304
変形済み表アクセス処理手段
2305
変形済み表
2306
変形データ H2
2307
変形済みデータ処理手段
2308
処理済変形データ H3
2309
変形済み第2撹乱用データ X2o
2313
データ変形処理手段
2310
処理済変形データ H4
2311
データ逆変形処理手段
2312
変形済みデータ処理手段
2317
処理済変形データ H5
2314
処理済変形データ H7
2318
変形済み表アクセス処理手段
2315
データ逆変形処理手段
2319
処理済変形データ H6
2316
処理済データ D2
2320

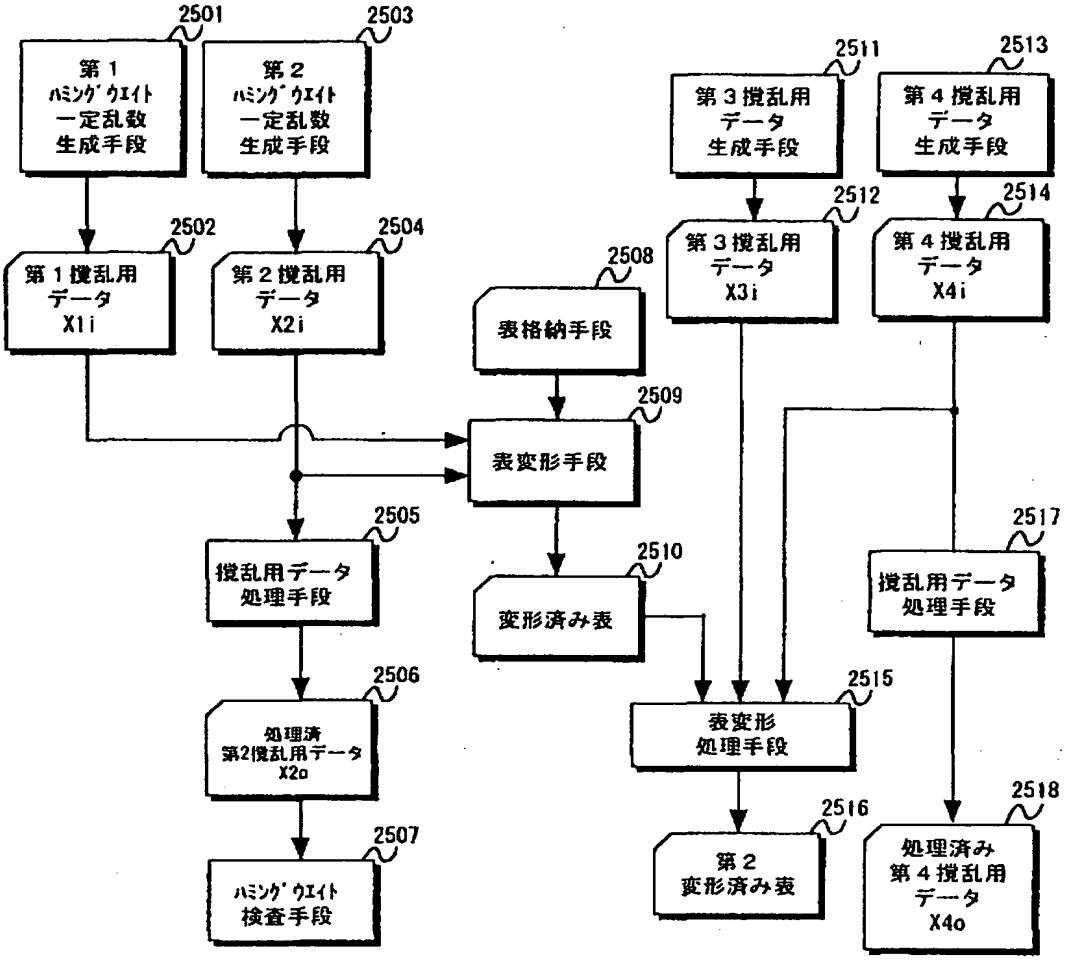
【図２４】

図 24


入力データ D1
2401
データ変形処理手段
2402
変形データ H1
2404
データ変形処理手段
2405
変形データ H2
2407
変形済み表アクセス処理手段
2408
変形データ H3
2410
変形済みデータ処理手段
2411
処理済変形データ H4
2412
データ変形処理手段
2413
処理済変形データ H5
2414
データ変形処理手段
2415
処理済変形データ H6
2416
第３撹乱用データ X3i
2403
第１撹乱用データ X1i
2406
第２変形済み表
2409
処理済み第２撹乱用データ X2o
2418
処理済み第４撹乱用データ X4o
2421
データ逆変形処理手段
2417
処理済変形データ H7
2419
データ逆変形処理手段
2420
処理済変形データ H8
2422
変形済み表アクセス処理手段
2423
変形データ H9
2424
変形済みデータ処理手段
2425
処理済変形データ H10
2426
データ逆変形処理手段
2427
処理済変形データ H11
2428
データ逆変形処理手段
2429
処理済データ D2
2430

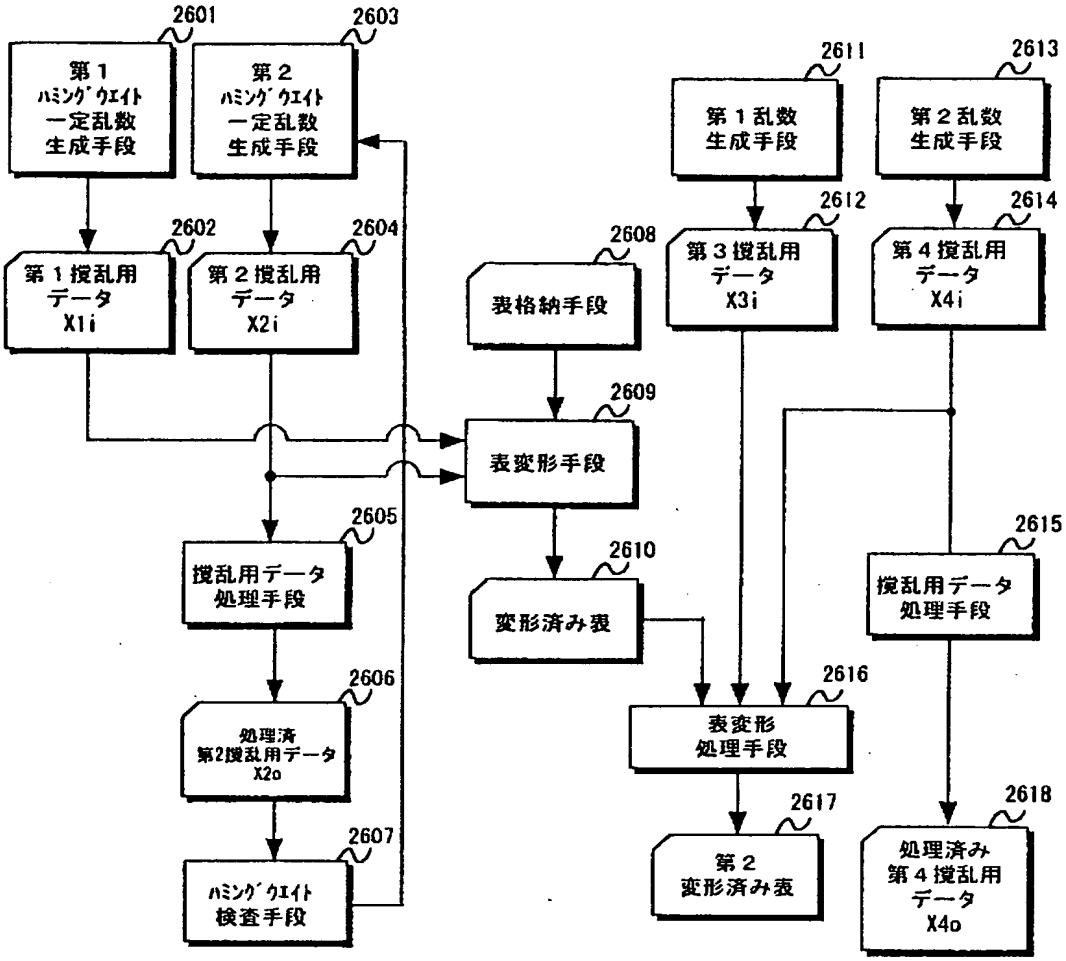
【図２５】

図 25


2501
第１
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
2503
第２
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
2511
第３撹乱用
データ
生成手段
2513
第４撹乱用
データ
生成手段
2502
第１撹乱用
データ
X1i
2504
第２撹乱用
データ
X2i
2508
表格納手段
2512
第３撹乱用
データ
X3i
2514
第４撹乱用
データ
X4i
2509
表変形手段
2505
撹乱用データ
処理手段
2510
変形済み表
2517
撹乱用データ
処理手段
2506
処理済
第2撹乱用データ
X2o
2515
表変形
処理手段
2507
ﾊﾐﾝｸﾞｳｴｲﾄ
検査手段
2516
第２
変形済み表
2518
処理済み
第４撹乱用
データ
X4o

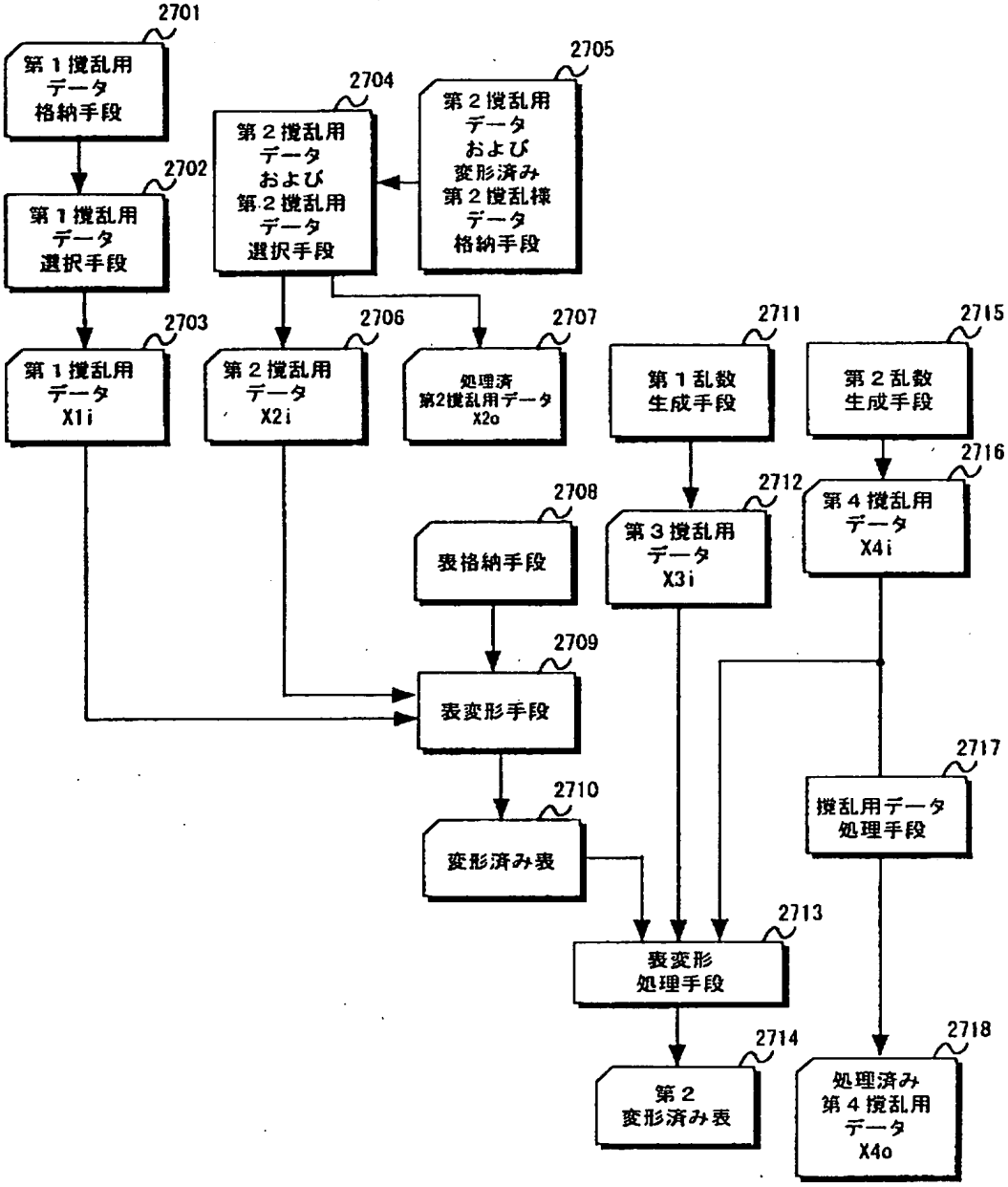
【図２６】

図　26


2601
第１
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
2603
第２
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
2611
第１乱数
生成手段
2613
第２乱数
生成手段
2602
第１撹乱用
データ
X1i
2604
第２撹乱用
データ
X2i
2608
表格納手段
2612
第３撹乱用
データ
X3i
2614
第４撹乱用
データ
X4i
2609
表変形手段
2605
撹乱用データ
処理手段
2610
変形済み表
2615
撹乱用データ
処理手段
2606
処理済
第2撹乱用データ
X2o
2616
表変形
処理手段
2607
ﾊﾐﾝｸﾞｳｴｲﾄ
検査手段
2617
第２
変形済み表
2618
処理済み
第４撹乱用
データ
X4o

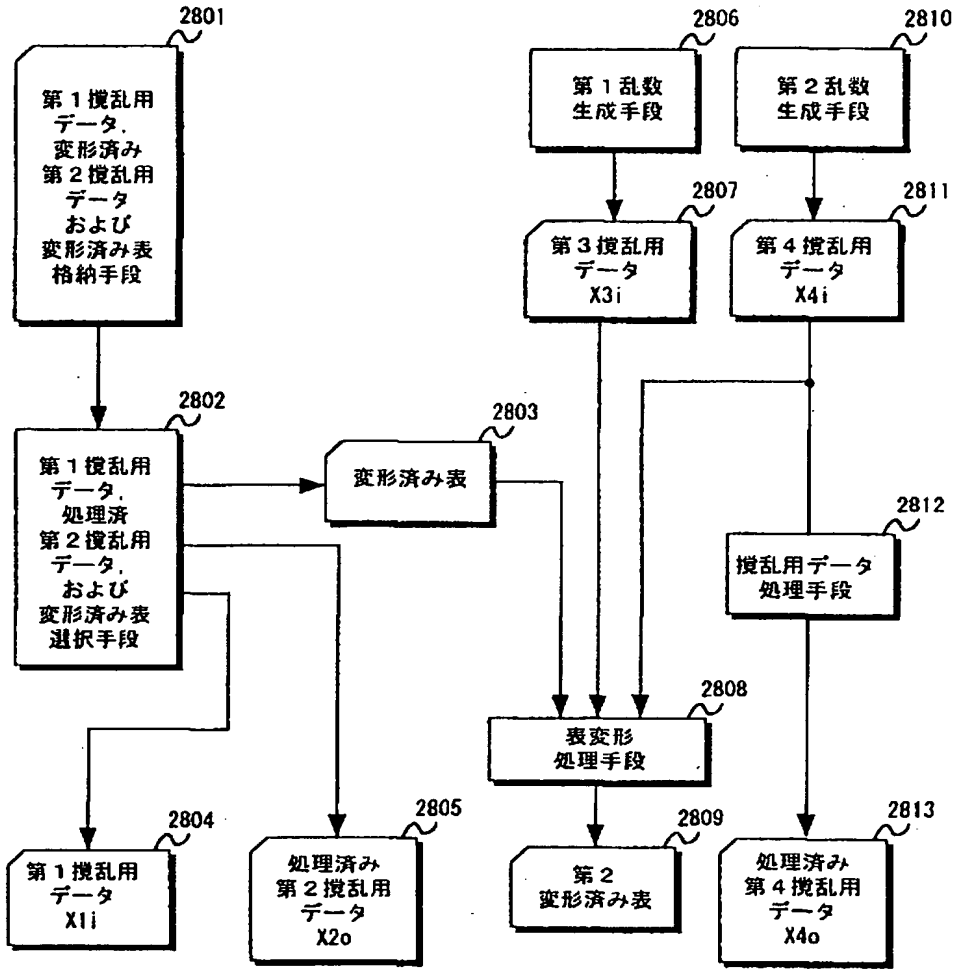
【図２７】

図　27


2701
第１撹乱用データ格納手段
2702
第１撹乱用データ選択手段
2703
第１撹乱用データX1i
2704
第２撹乱用データおよび第２撹乱用データ選択手段
2705
第２撹乱用データおよび変形済み第２撹乱様データ格納手段
2706
第２撹乱用データX2i
2707
処理済第2撹乱用データX2o
2708
表格納手段
2709
表変形手段
2710
変形済み表
2711
第１乱数生成手段
2712
第３撹乱用データX3i
2713
表変形処理手段
2714
第２変形済み表
2715
第２乱数生成手段
2716
第４撹乱用データX4i
2717
撹乱用データ処理手段
2718
処理済み第４撹乱用データX4o

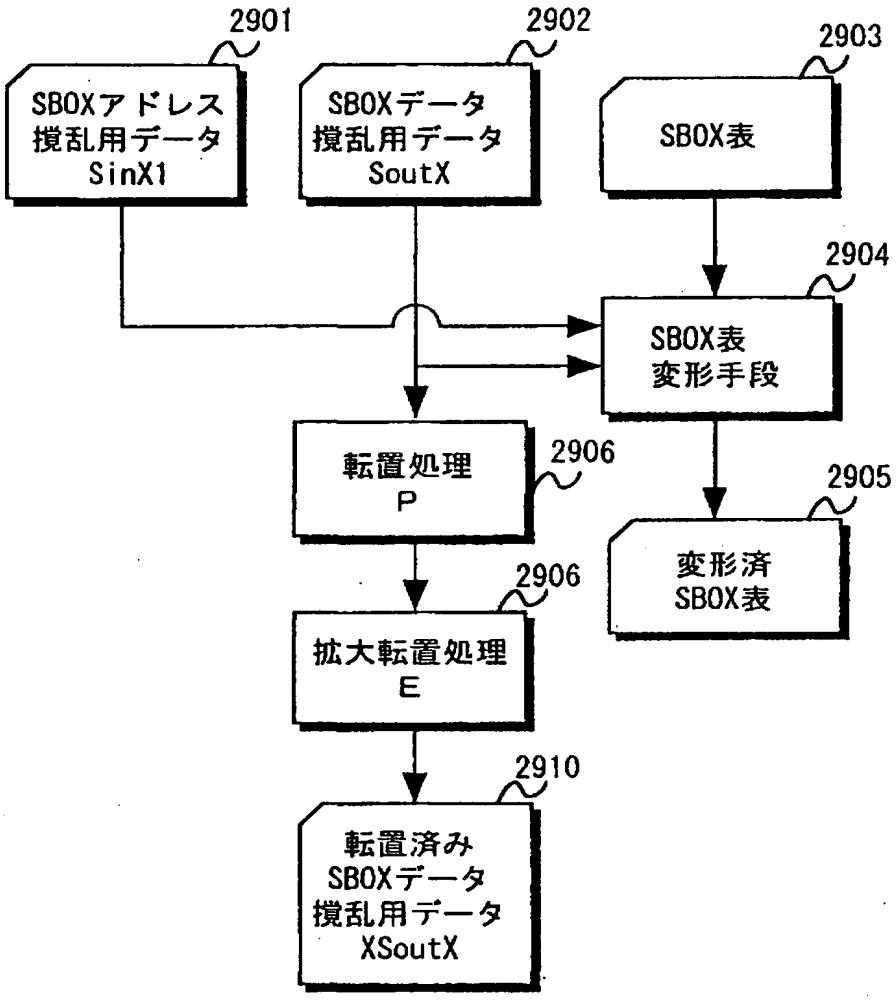
【図２８】

図　28


2801
第１撹乱用データ，変形済み第２撹乱用データおよび変形済み表格納手段
2806
第１乱数生成手段
2810
第２乱数生成手段
2807
第３撹乱用データ X3i
2811
第４撹乱用データ X4i
2802
第１撹乱用データ，処理済第２撹乱用データ，および変形済み表選択手段
2803
変形済み表
2812
撹乱用データ処理手段
2808
表変形処理手段
2804
第１撹乱用データ X1i
2805
処理済み第２撹乱用データ X2o
2809
第２変形済み表
2813
処理済み第４撹乱用データ X4o

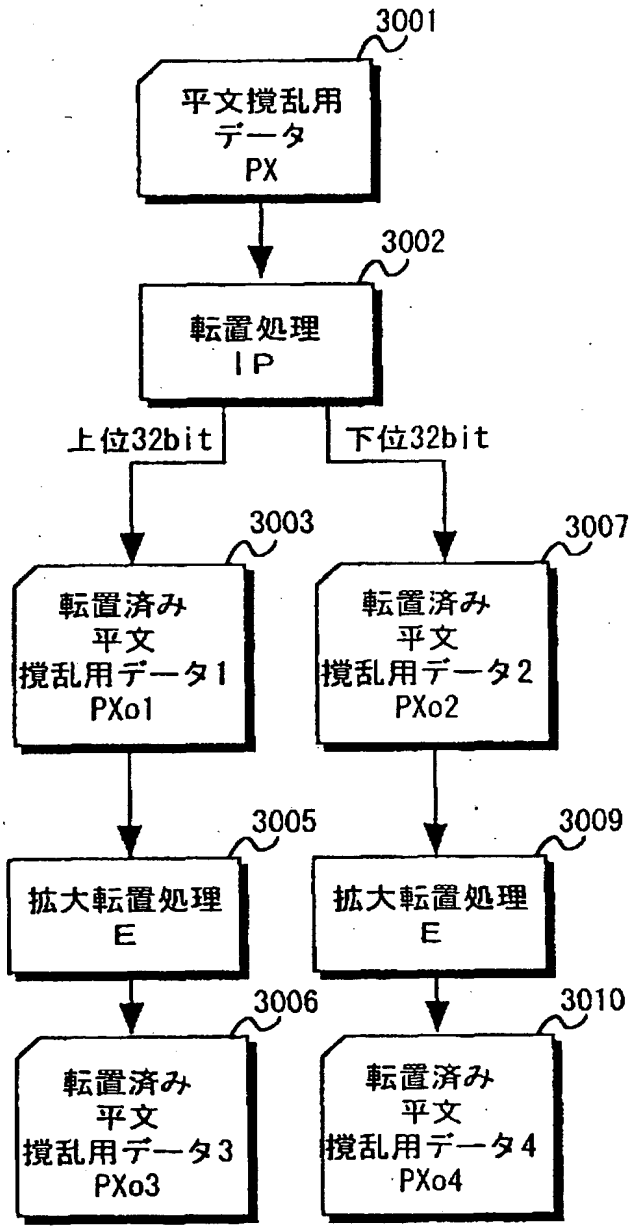
【図２９】

図 29


2901
SBOXアドレス
撹乱用データ
SinX1
2902
SBOXデータ
撹乱用データ
SoutX
2903
SBOX表
2904
SBOX表
変形手段
転置処理
P
2906
2905
変形済
SBOX表
2906
拡大転置処理
E
2910
転置済み
SBOXデータ
撹乱用データ
XSoutX

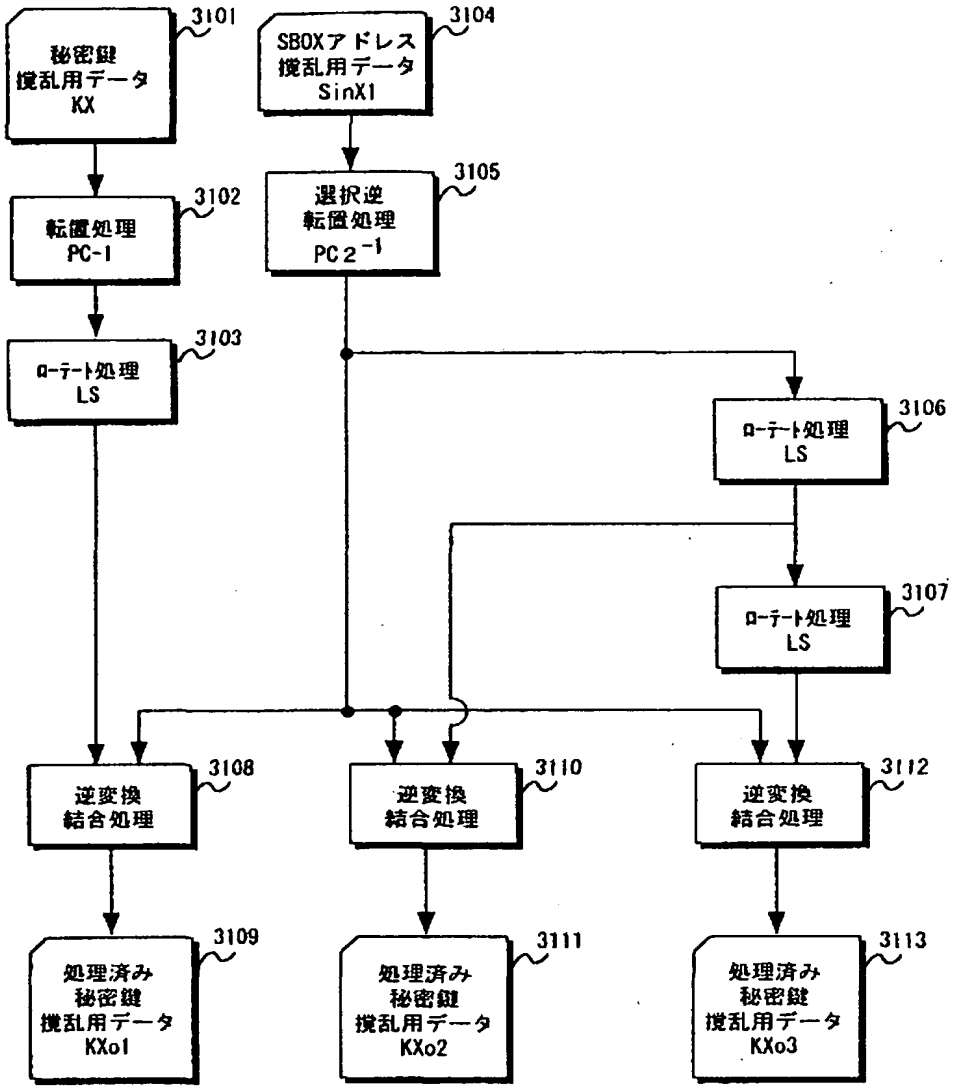
【図３０】

図　30


3001
平文撹乱用
データ
PX
3002
転置処理
IP
上位32bit
下位32bit
3003
転置済み
平文
撹乱用データ1
PXo1
3007
転置済み
平文
撹乱用データ2
PXo2
3005
拡大転置処理
E
3009
拡大転置処理
E
3006
転置済み
平文
撹乱用データ3
PXo3
3010
転置済み
平文
撹乱用データ4
PXo4

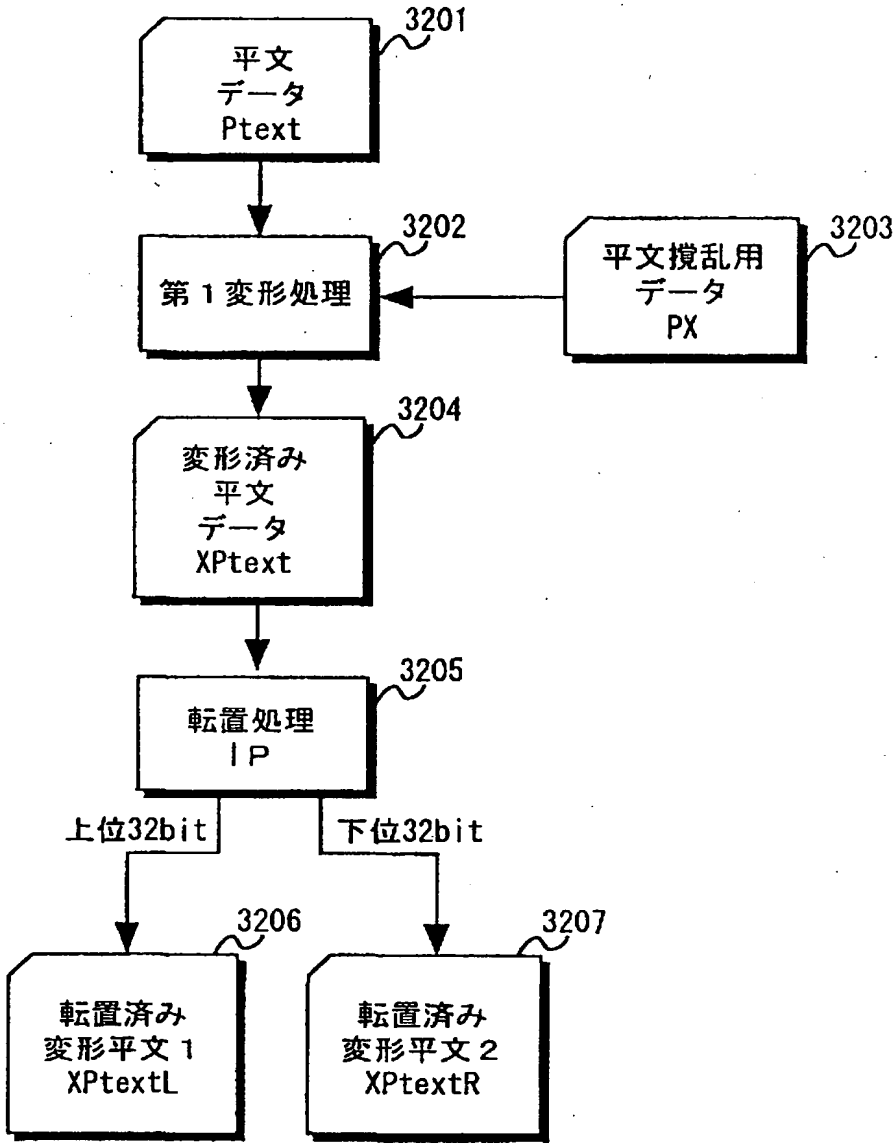
【図３１】

図 31


秘密鍵
撹乱用データ
KX
3101
SBOXアドレス
撹乱用データ
SinX1
3104
転置処理
PC-1
3102
選択逆
転置処理
PC2-1
3105
ローテート処理
LS
3103
ローテート処理
LS
3106
ローテート処理
LS
3107
逆変換
結合処理
3108
逆変換
結合処理
3110
逆変換
結合処理
3112
処理済み
秘密鍵
撹乱用データ
KXo1
3109
処理済み
秘密鍵
撹乱用データ
KXo2
3111
処理済み
秘密鍵
撹乱用データ
KXo3
3113

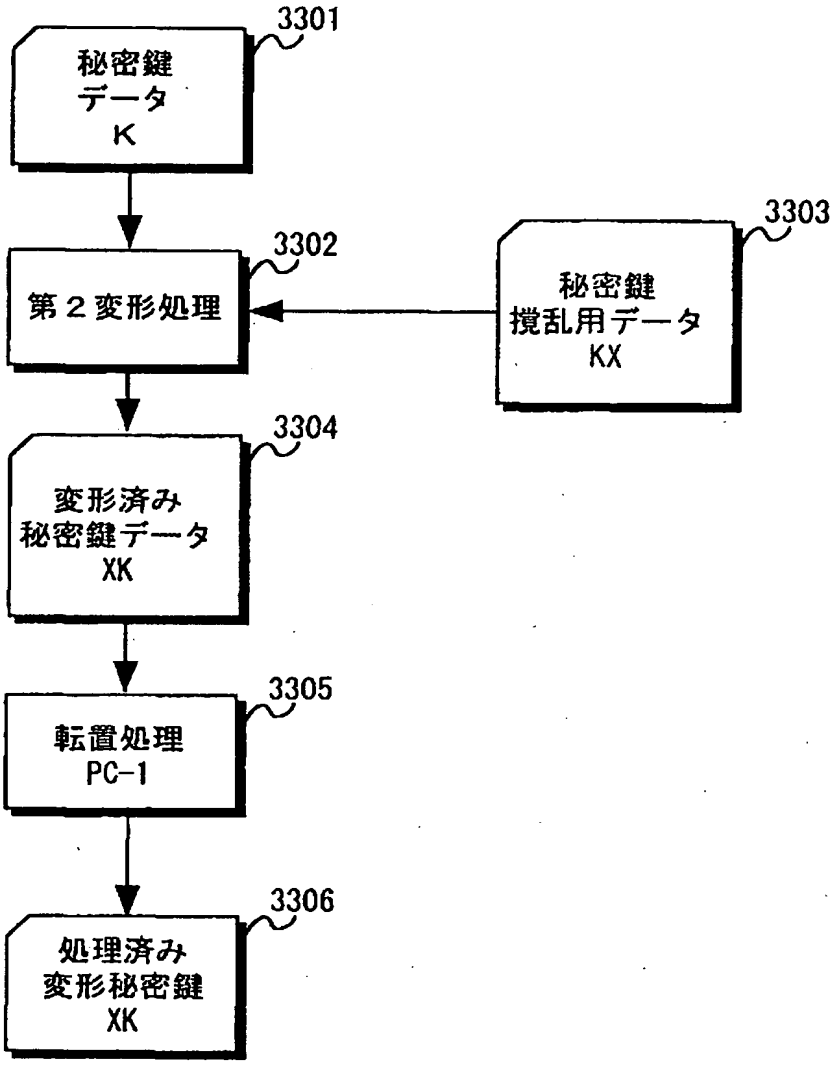
【図３２】

図 32


3201
平文
データ
Ptext
3202
第１変形処理
3203
平文撹乱用
データ
PX
3204
変形済み
平文
データ
XPtext
3205
転置処理
IP
上位32bit
下位32bit
3206
転置済み
変形平文１
XPtextL
3207
転置済み
変形平文２
XPtextR

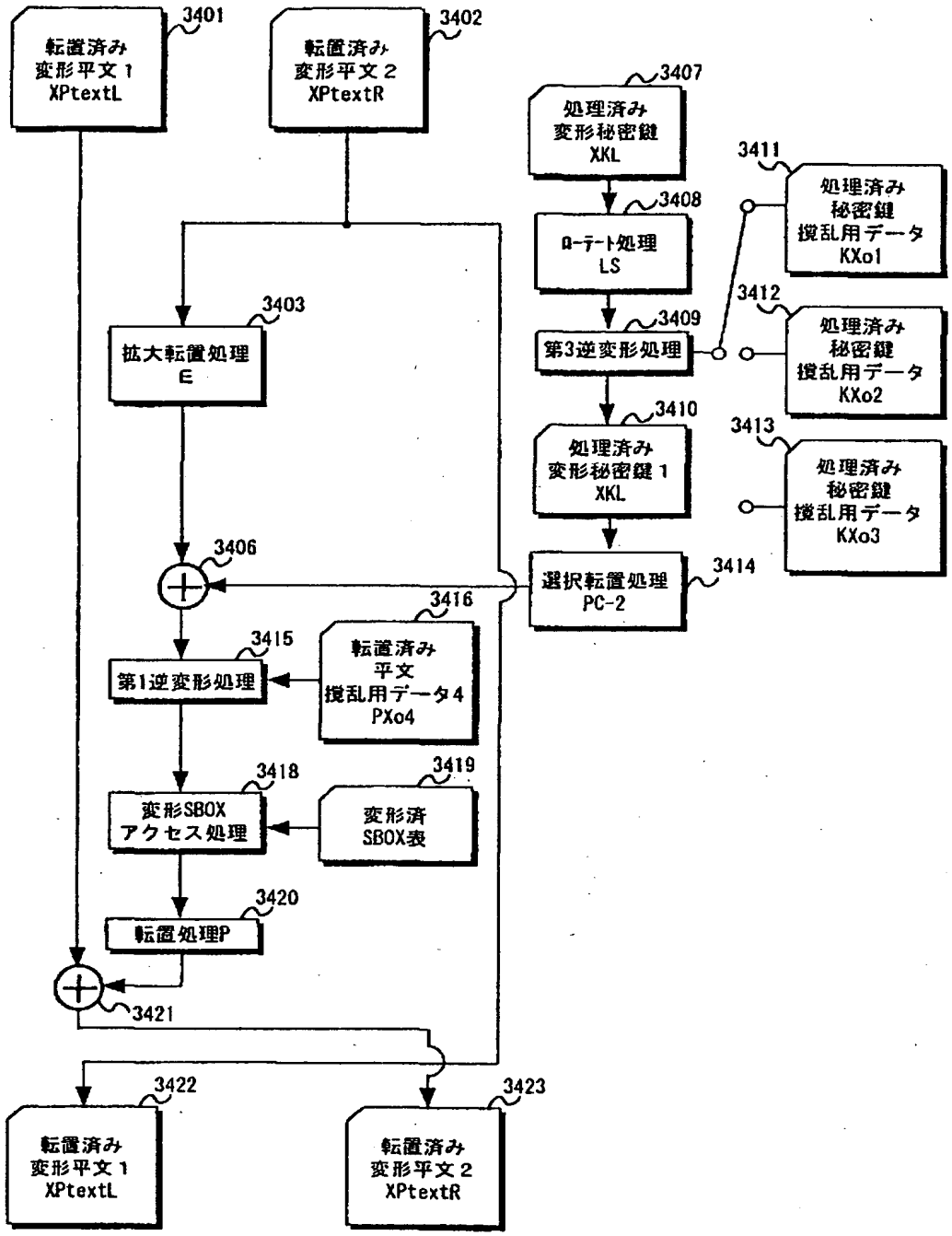
【図３３】

図 33


3301
秘密鍵
データ
K
3302
第２変形処理
3303
秘密鍵
撹乱用データ
KX
3304
変形済み
秘密鍵データ
XK
3305
転置処理
PC-1
3306
処理済み
変形秘密鍵
XK

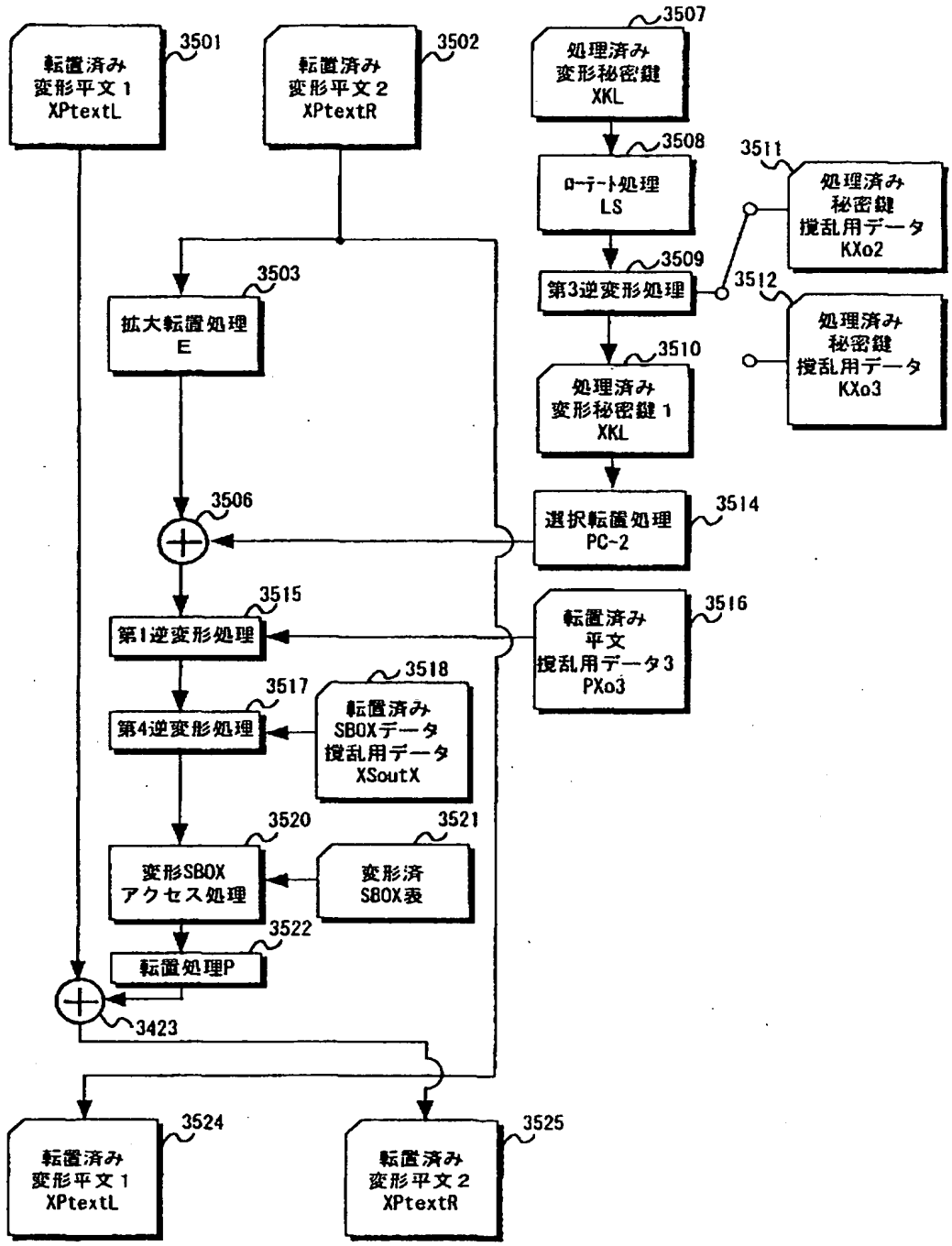
【図３４】

図 34


3401
転置済み
変形平文１
XPtextL
3402
転置済み
変形平文２
XPtextR
3407
処理済み
変形秘密鍵
XKL
3411
処理済み
秘密鍵
撹乱用データ
KXo1
3408
ローテート処理
LS
3403
拡大転置処理
E
3412
処理済み
秘密鍵
撹乱用データ
KXo2
3409
第3逆変形処理
3410
処理済み
変形秘密鍵１
XKL
3413
処理済み
秘密鍵
撹乱用データ
KXo3
3406
3414
選択転置処理
PC-2
3416
転置済み
平文
撹乱用データ４
PXo4
3415
第1逆変形処理
3418
変形SBOX
アクセス処理
3419
変形済
SBOX表
3420
転置処理P
3421
3422
転置済み
変形平文１
XPtextL
3423
転置済み
変形平文２
XPtextR

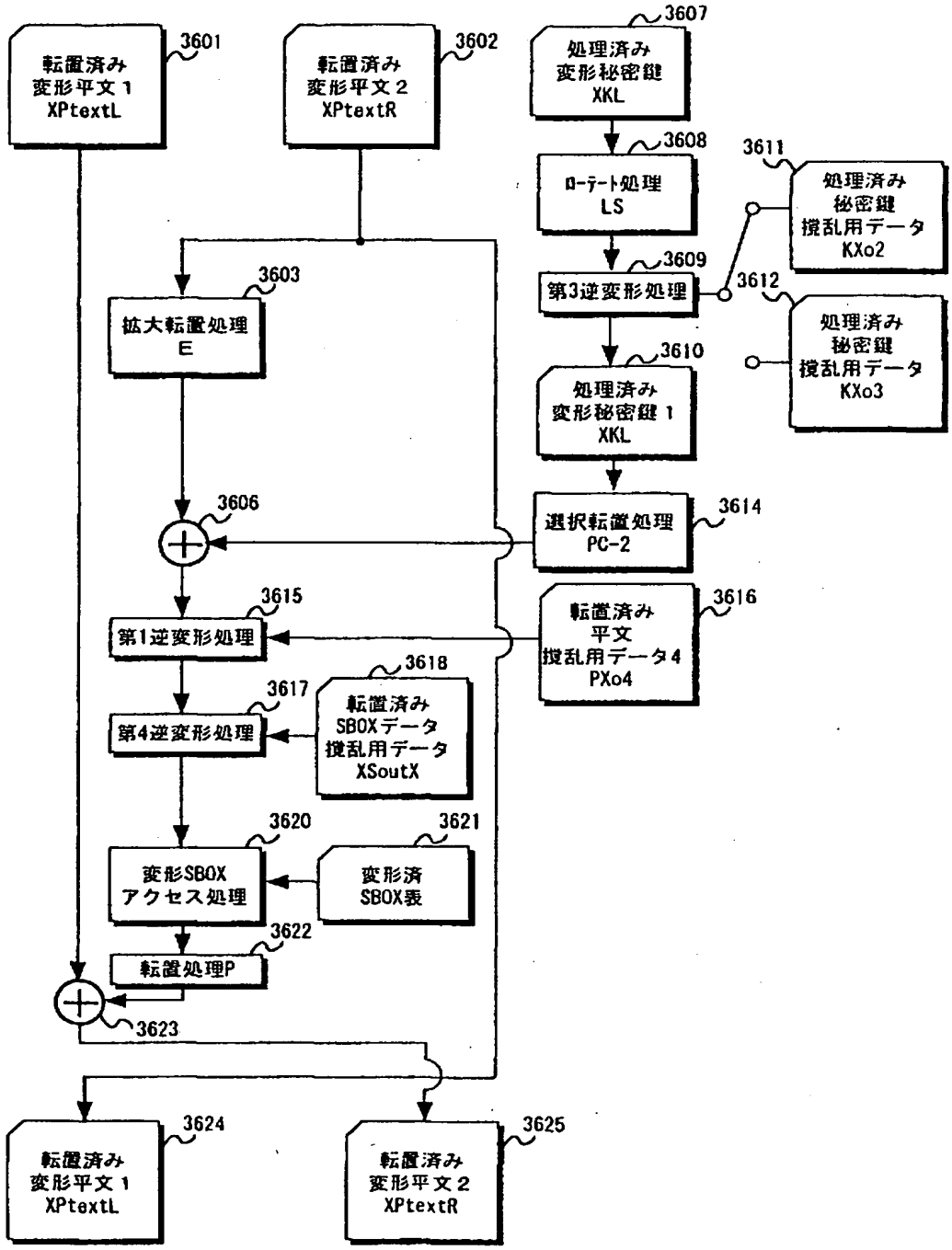
【図３５】

図 35


転置済み
変形平文１
XPtextL
3501
転置済み
変形平文２
XPtextR
3502
処理済み
変形秘密鍵
XKL
3507
ローテート処理
LS
3508
処理済み
秘密鍵
撹乱用データ
KXo2
3511
第3逆変形処理
3509
3512
処理済み
秘密鍵
撹乱用データ
KXo3
拡大転置処理
E
3503
処理済み
変形秘密鍵１
XKL
3510
3506
選択転置処理
PC-2
3514
第1逆変形処理
3515
転置済み
平文
撹乱用データ３
PXo3
3516
第4逆変形処理
3517
転置済み
SBOXデータ
撹乱用データ
XSoutX
3518
変形SBOX
アクセス処理
3520
変形済
SBOX表
3521
転置処理P
3522
3423
転置済み
変形平文１
XPtextL
3524
転置済み
変形平文２
XPtextR
3525

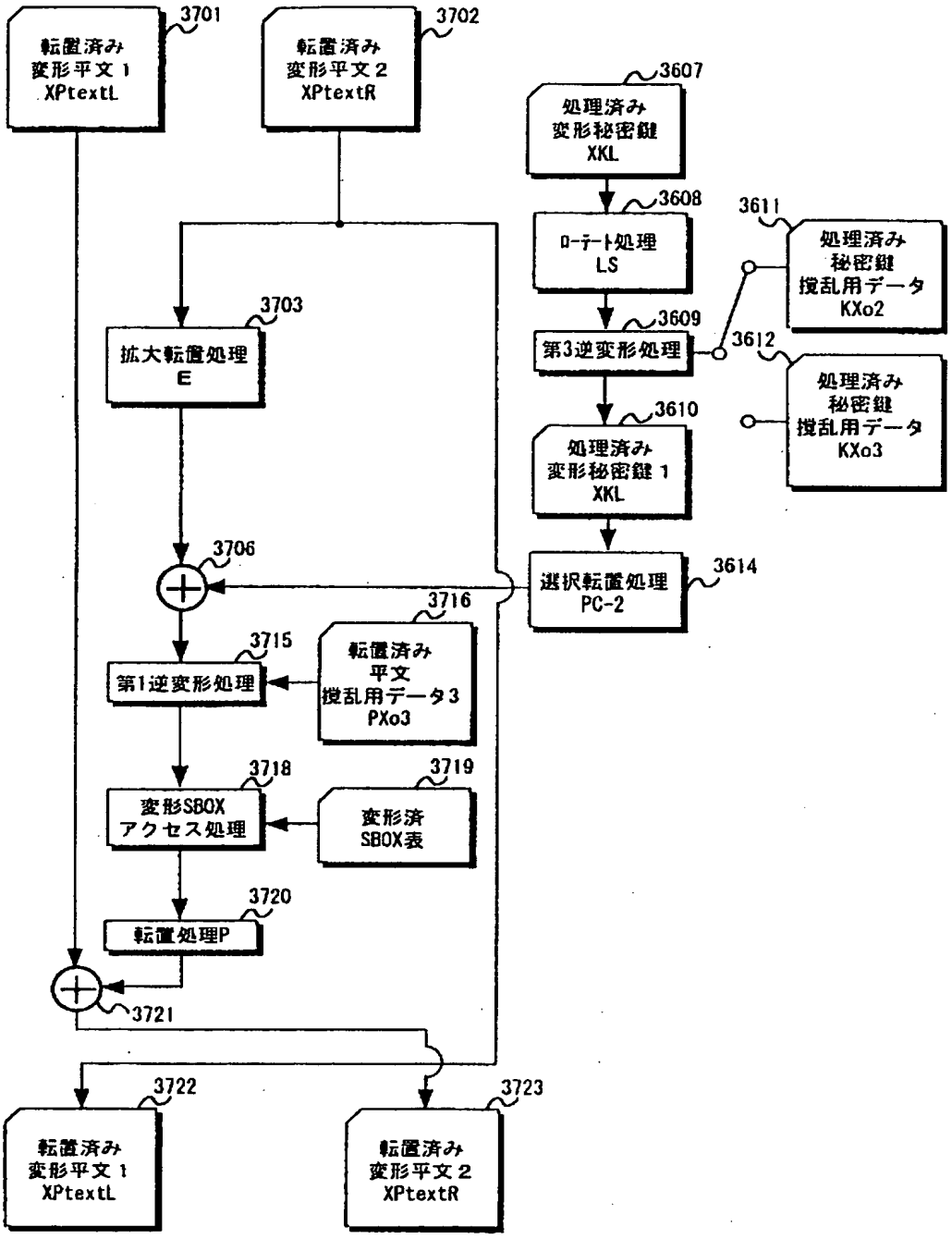
【図３６】

図　36


転置済み
変形平文１
XPtextL
3601
転置済み
変形平文２
XPtextR
3602
処理済み
変形秘密鍵
XKL
3607
ローテート処理
LS
3608
処理済み
秘密鍵
撹乱用データ
KXo2
3611
第3逆変形処理
3609
3612
処理済み
秘密鍵
撹乱用データ
KXo3
拡大転置処理
E
3603
処理済み
変形秘密鍵１
XKL
3610
選択転置処理
PC-2
3614
3606
転置済み
平文
撹乱用データ４
PXo4
3616
第1逆変形処理
3615
転置済み
SBOXデータ
撹乱用データ
XSoutX
3618
第4逆変形処理
3617
変形SBOX
アクセス処理
3620
変形済
SBOX表
3621
転置処理P
3622
3623
転置済み
変形平文１
XPtextL
3624
転置済み
変形平文２
XPtextR
3625

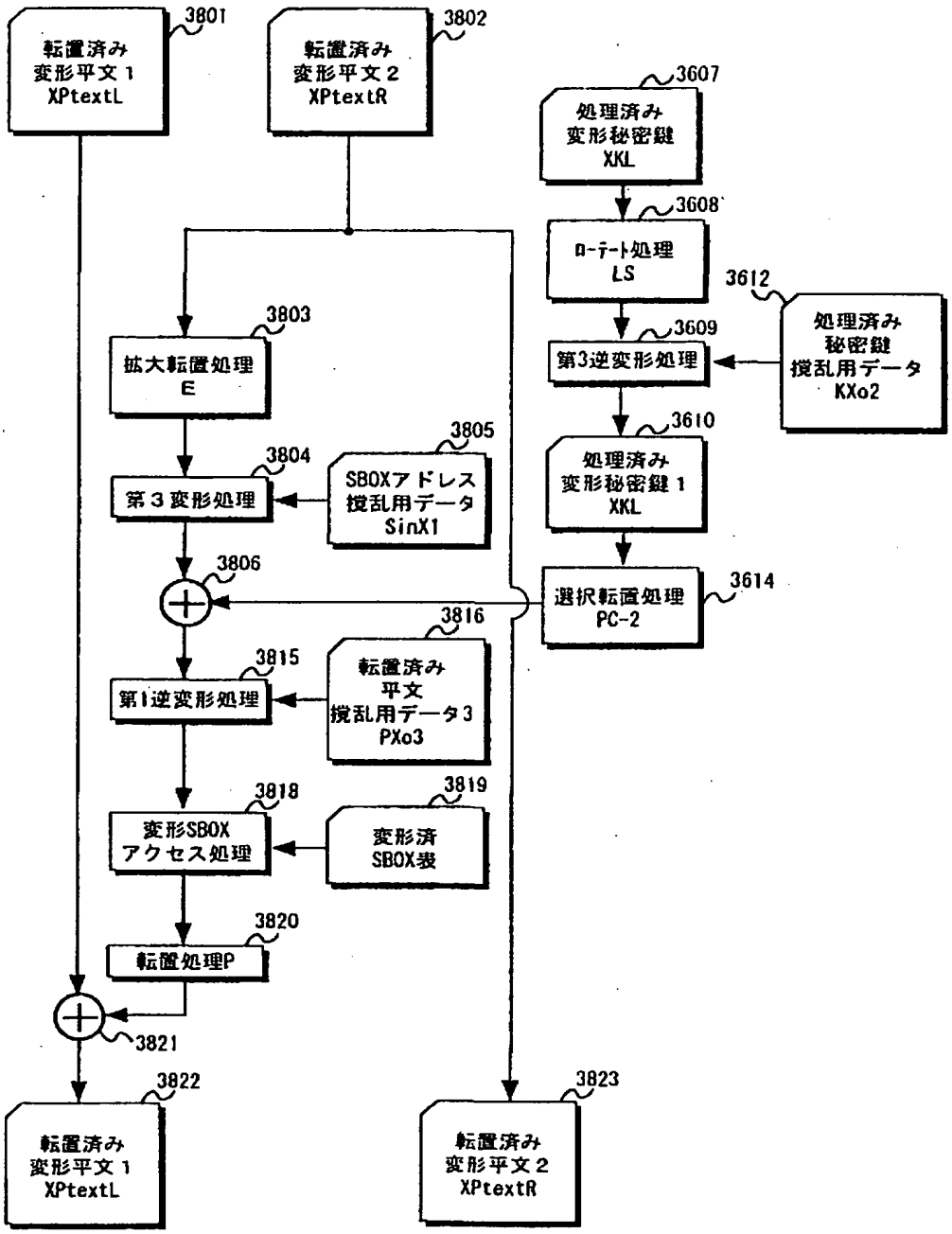
【図３７】

図　37


転置済み
変形平文１
XPtextL
3701
転置済み
変形平文２
XPtextR
3702
処理済み
変形秘密鍵
XKL
3607
ローテート処理
LS
3608
処理済み
秘密鍵
撹乱用データ
KXo2
3611
第3逆変形処理
3609
3612
処理済み
秘密鍵
撹乱用データ
KXo3
処理済み
変形秘密鍵１
XKL
3610
拡大転置処理
E
3703
選択転置処理
PC-2
3614
3706
3716
転置済み
平文
撹乱用データ３
PXo3
第1逆変形処理
3715
変形SBOX
アクセス処理
3718
変形済
SBOX表
3719
転置処理P
3720
3721
転置済み
変形平文１
XPtextL
3722
転置済み
変形平文２
XPtextR
3723

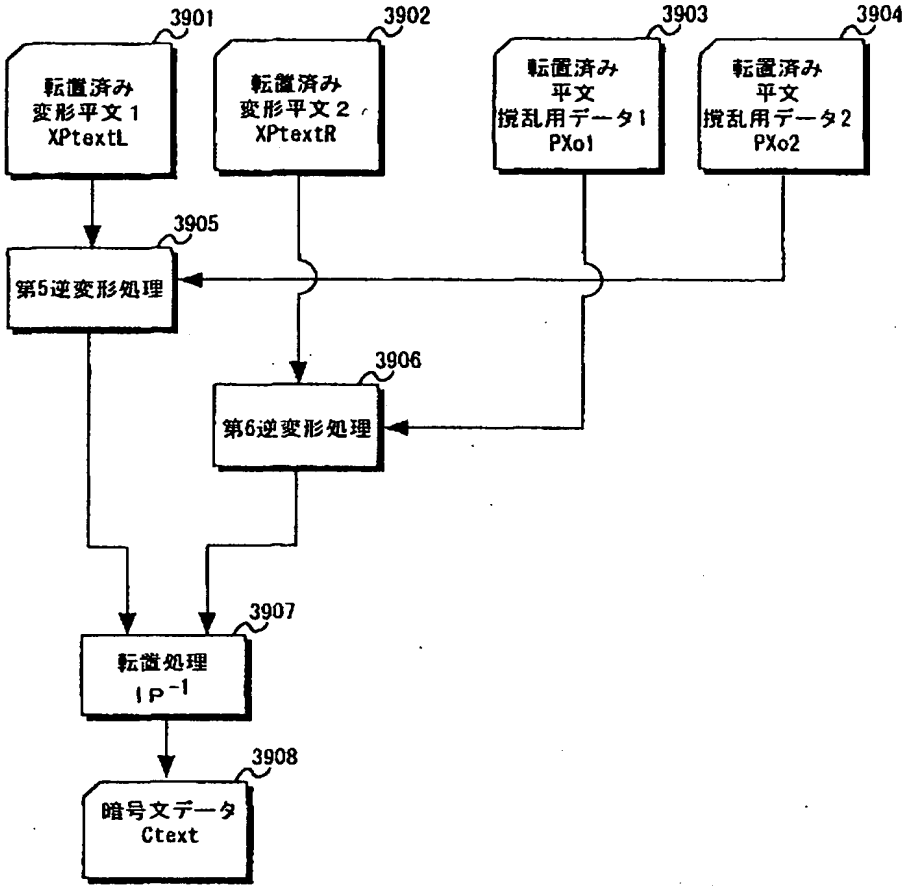
【図３８】

図 38


3801
転置済み
変形平文１
XPtextL
3802
転置済み
変形平文２
XPtextR
3607
処理済み
変形秘密鍵
XKL
3608
ローテート処理
LS
3612
処理済み
秘密鍵
撹乱用データ
KXo2
3609
第3逆変形処理
3610
処理済み
変形秘密鍵１
XKL
3614
選択転置処理
PC-2
3803
拡大転置処理
E
3805
SBOXアドレス
撹乱用データ
SinX1
3804
第３変形処理
3806
3816
転置済み
平文
撹乱用データ３
PXo3
3815
第1逆変形処理
3818
変形SBOX
アクセス処理
3819
変形済
SBOX表
3820
転置処理P
3821
3822
転置済み
変形平文１
XPtextL
3823
転置済み
変形平文２
XPtextR

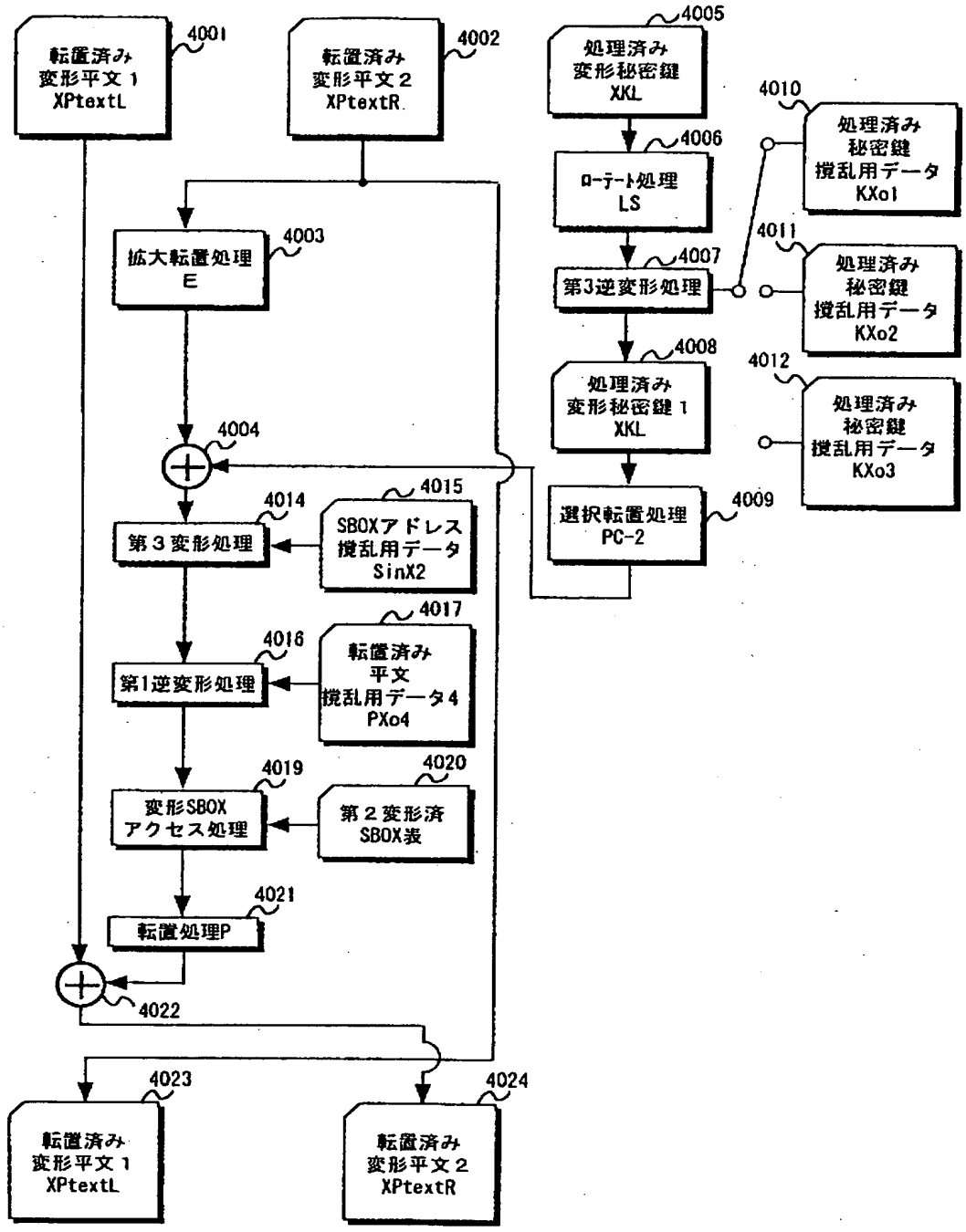
【図３９】

図 39


3901
転置済み
変形平文１
XPtextL
3902
転置済み
変形平文２
XPtextR
3903
転置済み
平文
撹乱用データ1
PXo1
3904
転置済み
平文
撹乱用データ2
PXo2
3905
第5逆変形処理
3906
第6逆変形処理
3907
転置処理
IP-1
3908
暗号文データ
Ctext

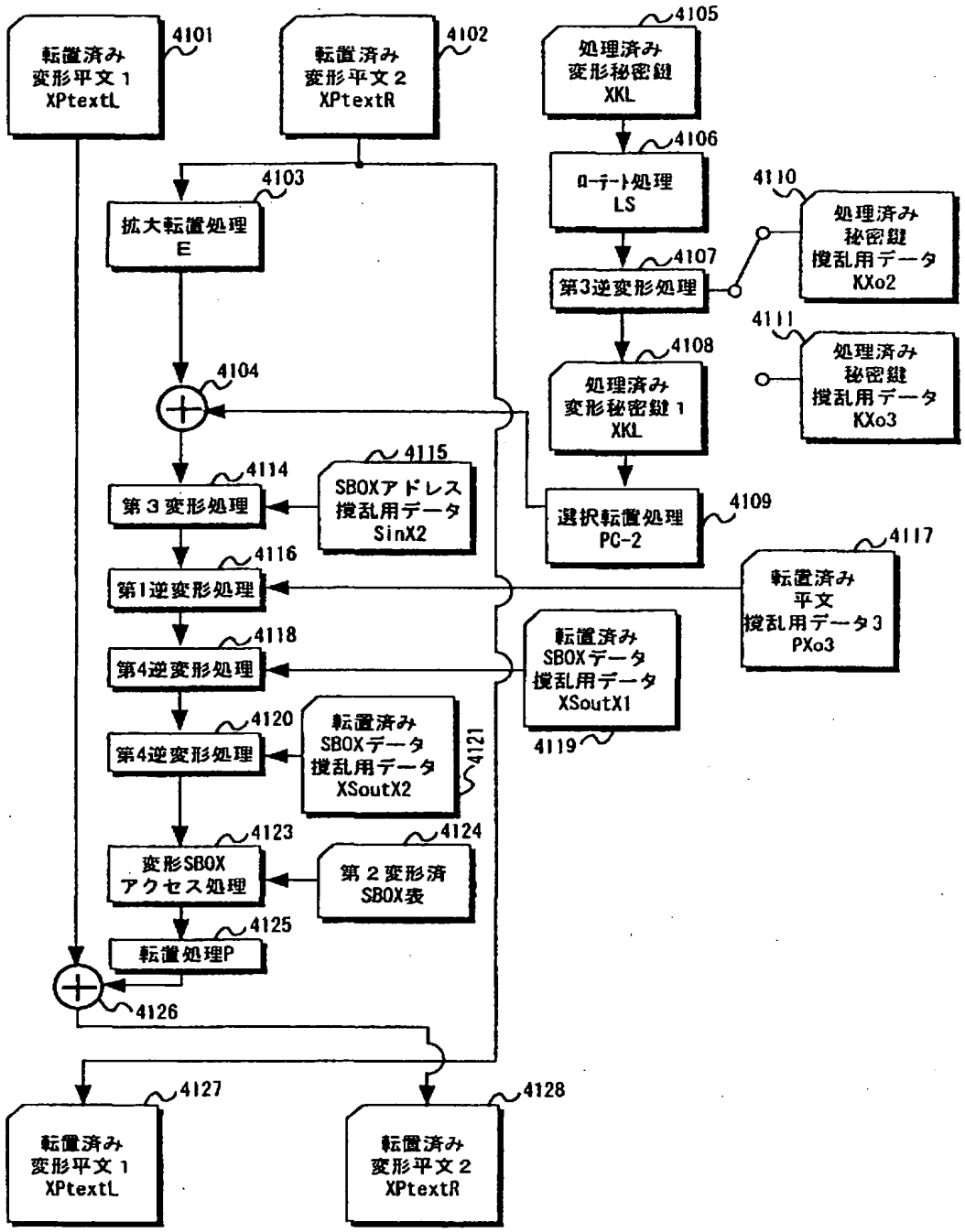
【図４０】

図　40


転置済み
変形平文１
XPtextL
4001
転置済み
変形平文２
XPtextR
4002
処理済み
変形秘密鍵
XKL
4005
4010
処理済み
秘密鍵
撹乱用データ
KXo1
ローテート処理
LS
4006
拡大転置処理
E
4003
第3逆変形処理
4007
4011
処理済み
秘密鍵
撹乱用データ
KXo2
4008
処理済み
変形秘密鍵１
XKL
4012
処理済み
秘密鍵
撹乱用データ
KXo3
4004
選択転置処理
PC-2
4009
4014
第３変形処理
4015
SBOXアドレス
撹乱用データ
SinX2
4017
転置済み
平文
撹乱用データ４
PXo4
4016
第1逆変形処理
4019
変形SBOX
アクセス処理
4020
第２変形済
SBOX表
4021
転置処理P
4022
4023
転置済み
変形平文１
XPtextL
4024
転置済み
変形平文２
XPtextR

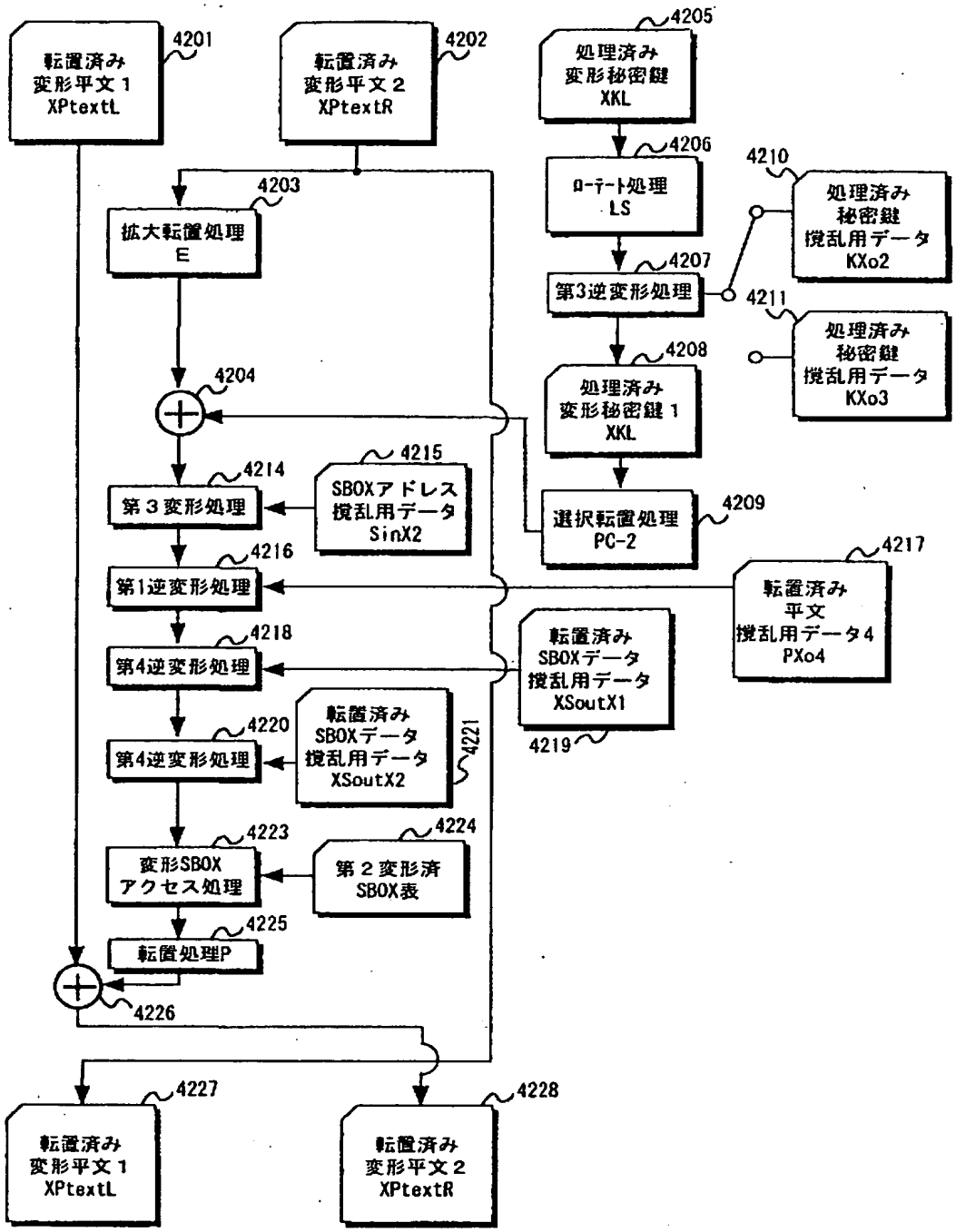
【図４１】

図 41


転置済み
変形平文１
XPtextL
4101
転置済み
変形平文２
XPtextR
4102
処理済み
変形秘密鍵
XKL
4105
ローテート処理
LS
4106
処理済み
秘密鍵
撹乱用データ
KXo2
4110
第3逆変形処理
4107
処理済み
秘密鍵
撹乱用データ
KXo3
4111
処理済み
変形秘密鍵１
XKL
4108
拡大転置処理
E
4103
4104
SBOXアドレス
撹乱用データ
SinX2
4115
第３変形処理
4114
選択転置処理
PC-2
4109
転置済み
平文
撹乱用データ３
PXo3
4117
第1逆変形処理
4116
転置済み
SBOXデータ
撹乱用データ
XSoutX1
4119
第4逆変形処理
4118
転置済み
SBOXデータ
撹乱用データ
XSoutX2
4121
第4逆変形処理
4120
第２変形済
SBOX表
4124
変形SBOX
アクセス処理
4123
転置処理P
4125
4126
転置済み
変形平文１
XPtextL
4127
転置済み
変形平文２
XPtextR
4128

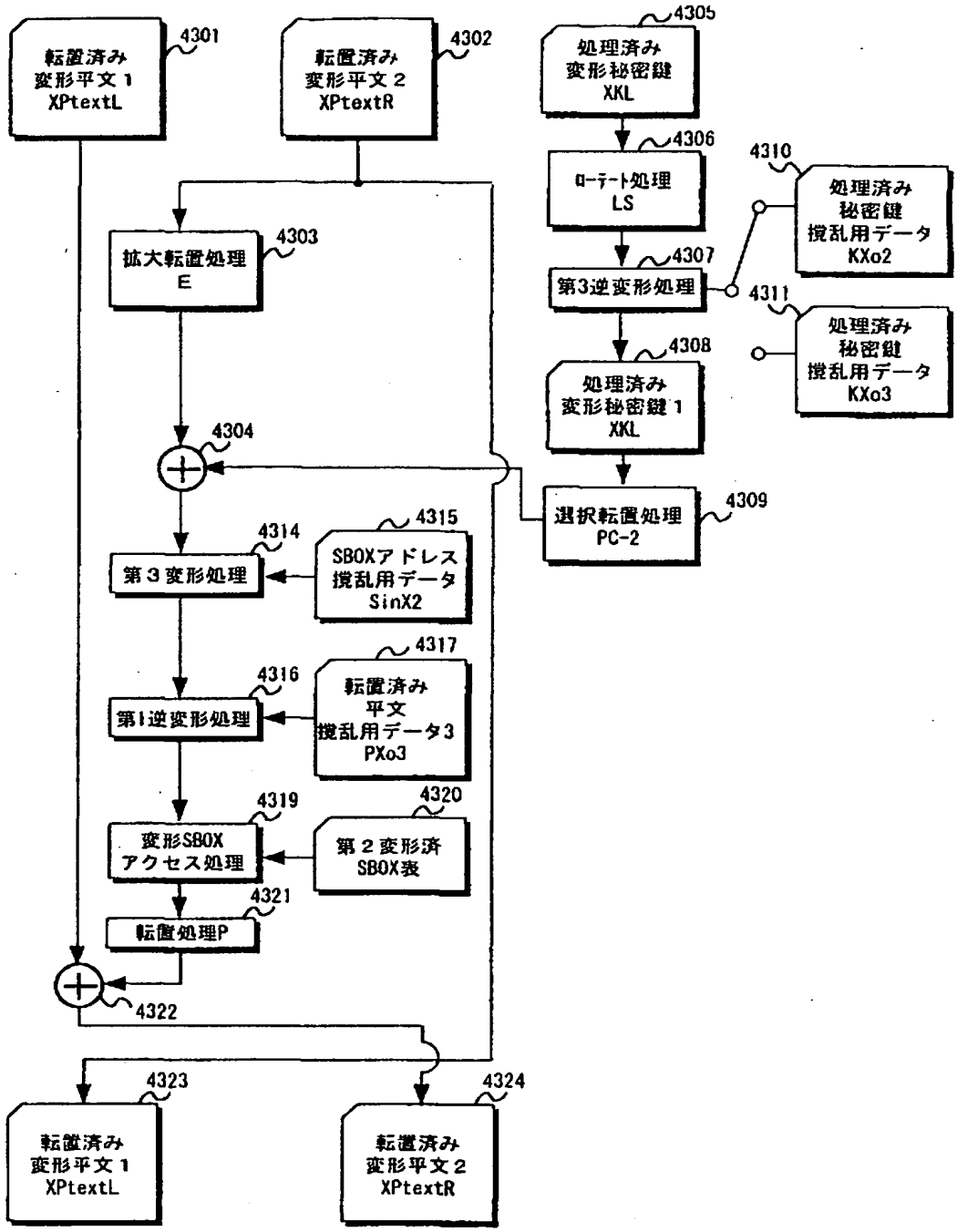
【図４２】

図 42


転置済み
変形平文１
XPtextL
4201
転置済み
変形平文２
XPtextR
4202
処理済み
変形秘密鍵
XKL
4205
ローテート処理
LS
4206
第3逆変形処理
4207
処理済み
変形秘密鍵１
XKL
4208
選択転置処理
PC-2
4209
処理済み
秘密鍵
撹乱用データ
KXo2
4210
処理済み
秘密鍵
撹乱用データ
KXo3
4211
拡大転置処理
E
4203
4204
第３変形処理
4214
SBOXアドレス
撹乱用データ
SinX2
4215
第1逆変形処理
4216
転置済み
平文
撹乱用データ４
PXo4
4217
第4逆変形処理
4218
転置済み
SBOXデータ
撹乱用データ
XSoutX1
4219
第4逆変形処理
4220
転置済み
SBOXデータ
撹乱用データ
XSoutX2
4221
変形SBOX
アクセス処理
4223
第２変形済
SBOX表
4224
転置処理P
4225
4226
転置済み
変形平文１
XPtextL
4227
転置済み
変形平文２
XPtextR
4228

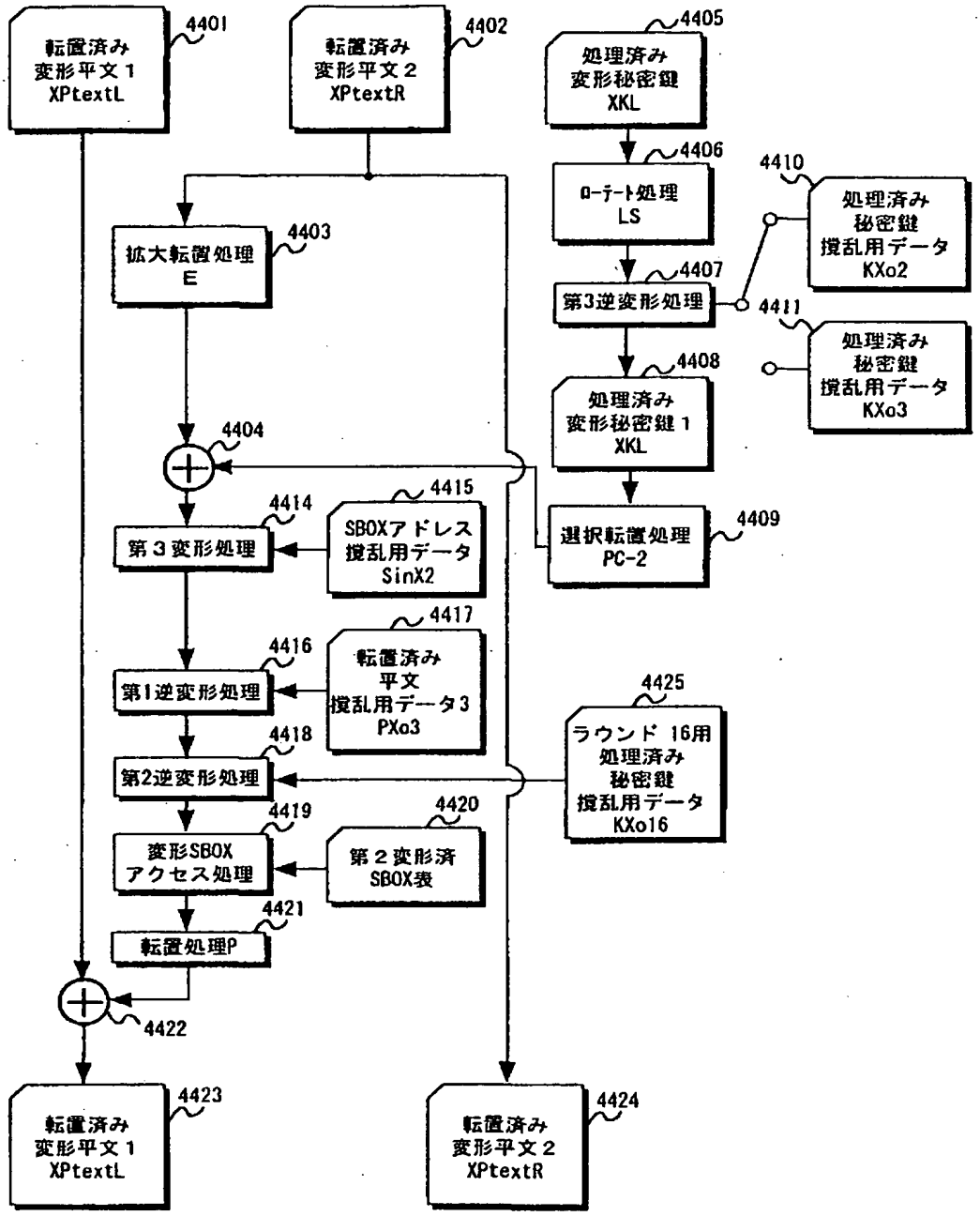
【図４３】

図 43


転置済み
変形平文１
XPtextL
4301
転置済み
変形平文２
XPtextR
4302
処理済み
変形秘密鍵
XKL
4305
ローテート処理
LS
4306
4310
処理済み
秘密鍵
撹乱用データ
KXo2
拡大転置処理
E
4303
第3逆変形処理
4307
4311
処理済み
秘密鍵
撹乱用データ
KXo3
4308
処理済み
変形秘密鍵１
XKL
4304
選択転置処理
PC-2
4309
4315
SBOXアドレス
撹乱用データ
SinX2
4314
第３変形処理
4317
転置済み
平文
撹乱用データ3
PXo3
4316
第1逆変形処理
4319
4320
変形SBOX
アクセス処理
第２変形済
SBOX表
4321
転置処理P
4322
4323
転置済み
変形平文１
XPtextL
4324
転置済み
変形平文２
XPtextR

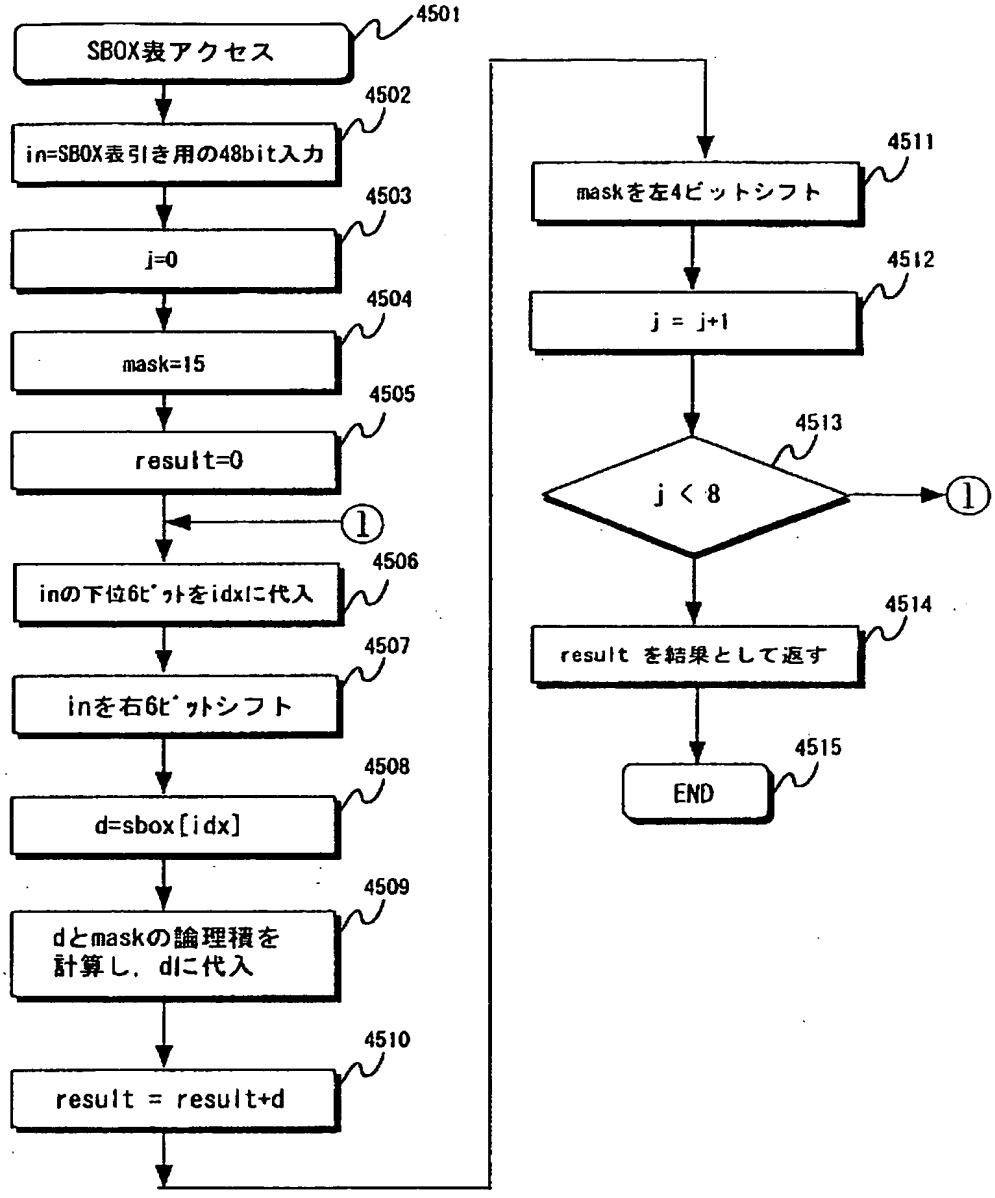
【図４４】

図 44


転置済み
変形平文１
XPtextL
4401
転置済み
変形平文２
XPtextR
4402
処理済み
変形秘密鍵
XKL
4405
ローテート処理
LS
4406
処理済み
秘密鍵
撹乱用データ
KXo2
4410
第3逆変形処理
4407
処理済み
秘密鍵
撹乱用データ
KXo3
4411
処理済み
変形秘密鍵１
XKL
4408
拡大転置処理
E
4403
4404
選択転置処理
PC-2
4409
第３変形処理
4414
SBOXアドレス
撹乱用データ
SinX2
4415
転置済み
平文
撹乱用データ３
PXo3
4417
第1逆変形処理
4416
ラウンド 16用
処理済み
秘密鍵
撹乱用データ
KXo16
4425
第2逆変形処理
4418
変形SBOX
アクセス処理
4419
第２変形済
SBOX表
4420
転置処理P
4421
4422
転置済み
変形平文１
XPtextL
4423
転置済み
変形平文２
XPtextR
4424

【図４５】

図 45


4501
SBOX表アクセス
4502
in=SBOX表引き用の48bit入力
4503
j=0
4504
mask=15
4505
result=0
①
4506
inの下位6ビットをidxに代入
4507
inを右6ビットシフト
4508
d=sbox[idx]
4509
dとmaskの論理積を
計算し，dに代入
4510
result = result+d
4511
maskを左4ビットシフト
4512
j = j+1
4513
j < 8
①
4514
result を結果として返す
4515
END

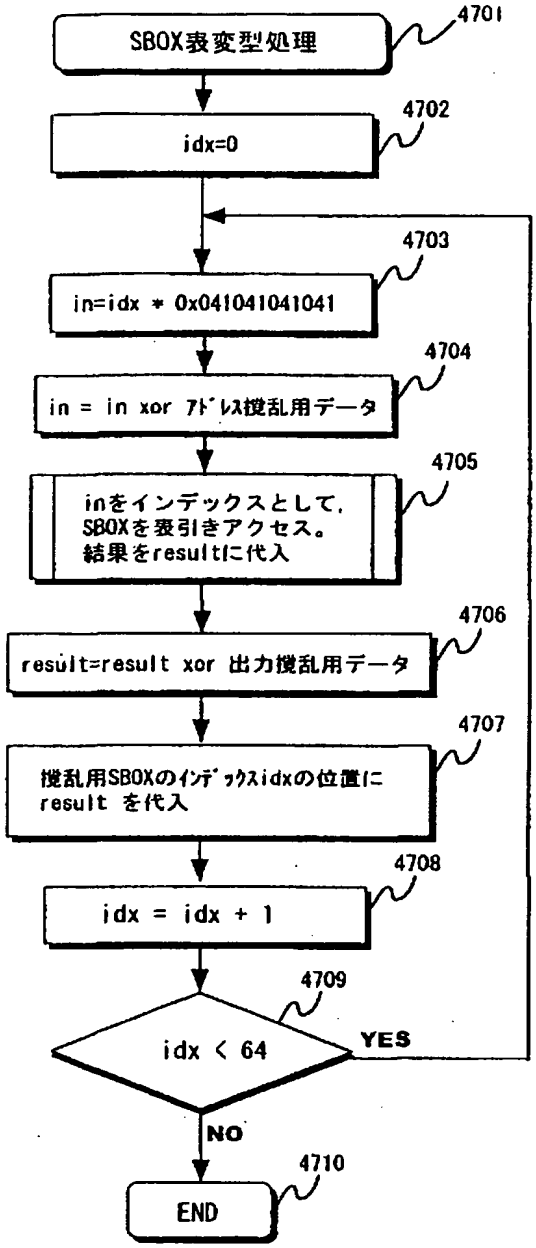
【図４６】

図　46

index	value	index	value	index	value	index	value
0	0xEFA72C4D	16	0x3911803A	32	0x40DA4917	48	0xF5BFF7A0
1	0x03DDEAD1	17	0xAC2456EC	33	0xFD13B462	49	0x5B496B9F
2	0x410DC1B2	18	0xA7D25DC9	34	0x1E662E4B	50	0xC81190F6
3	0xFD78BF0F	19	0x60870135	35	0xC8AF83B1	51	0xB6F4FE5C
4	0xD89E4A28	20	0x62C83393	36	0xE7491FB4	52	0x9C23C46A
5	0x740B24BD	21	0xC152FD56	37	0x8AD0C2DE	53	0x37E50109
6	0x1EE31FE4	22	0xCD75F47E	38	0x8B90B5D1	54	0x76CE5A8D
7	0x4795C278	23	0xBAECAECB	39	0x21067C87	55	0xEC3B97F0
8	0x266079F6	24	0x5CBBDE55	40	0xDA8CA2C9	56	0x3955610F
9	0xEF36474A	25	0x69C13020	41	0x436A1914	57	0xA0BCA6E3
10	0xFB36A20F	26	0x904C07A0	42	0x64FBD83C	58	0xA3A23D53
11	0x224F7C93	27	0x59BA9BFE	43	0x9F91E54A	59	0x05574025
12	0xB3F9B68B	28	0x0524E56C	44	0x2D377C7E	60	0x52E80B95
13	0xD860D917	29	0x3BFE8389	45	0x148D2FA8	61	0x6E225836
14	0x845A68D1	30	0x7A8F9B17	46	0xB10D83E2	62	0x0F74E628
15	0x1EA315A4	31	0x85196862	47	0x7278DA7D	63	0xD9CE3DCB

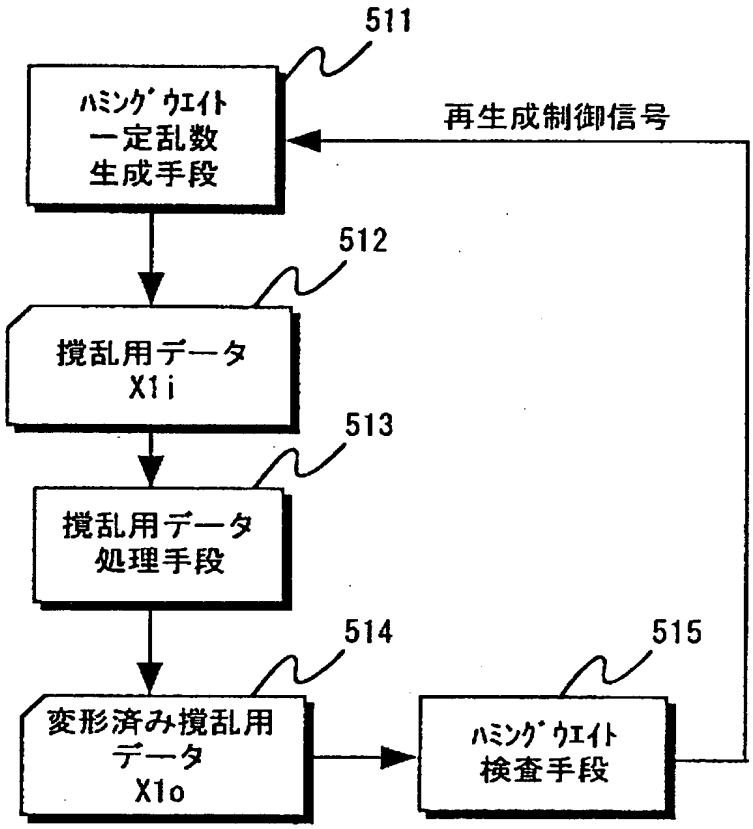
【図４７】

図　47


SBOX表変型処理
4701
idx=0
4702
in=idx * 0x041041041041
4703
in = in xor ｱﾄﾞﾚｽ撹乱用データ
4704
inをインデックスとして，
SBOXを表引きアクセス。
結果をresultに代入
4705
result=result xor 出力撹乱用データ
4706
撹乱用SBOXのｲﾝﾃﾞｯｸｽidxの位置に
result を代入
4707
idx = idx + 1
4708
idx < 64
4709
YES
NO
END
4710

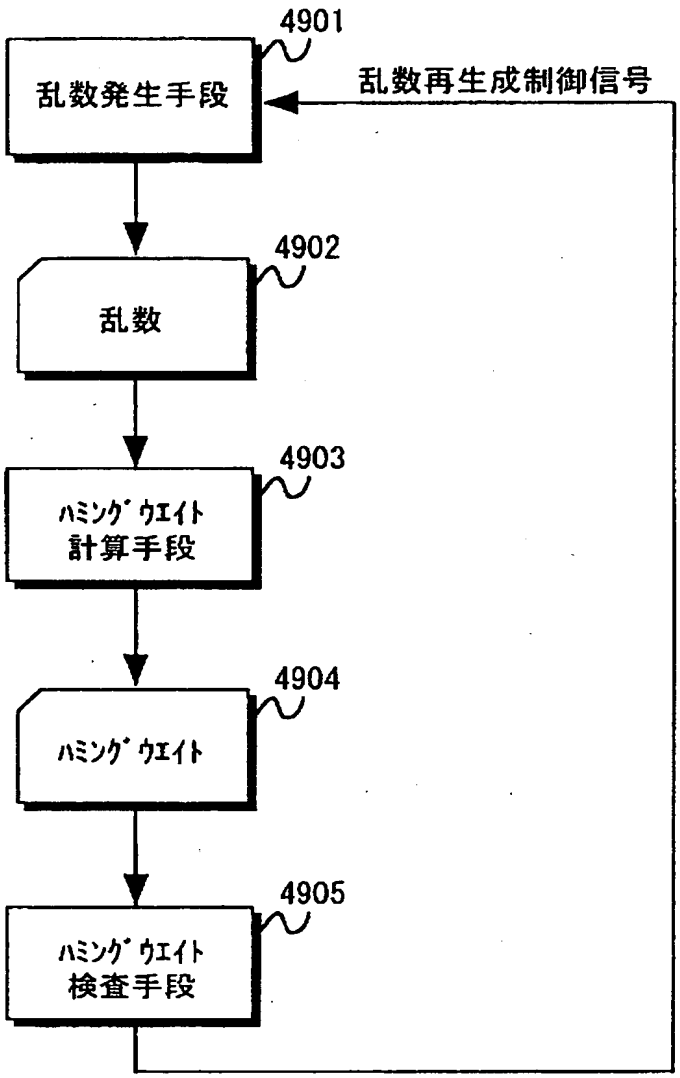
【図４８】

図　48


511
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数
生成手段
再生成制御信号
512
撹乱用データ
X1i
513
撹乱用データ
処理手段
514
変形済み撹乱用
データ
X1o
515
ﾊﾐﾝｸﾞｳｴｲﾄ
検査手段

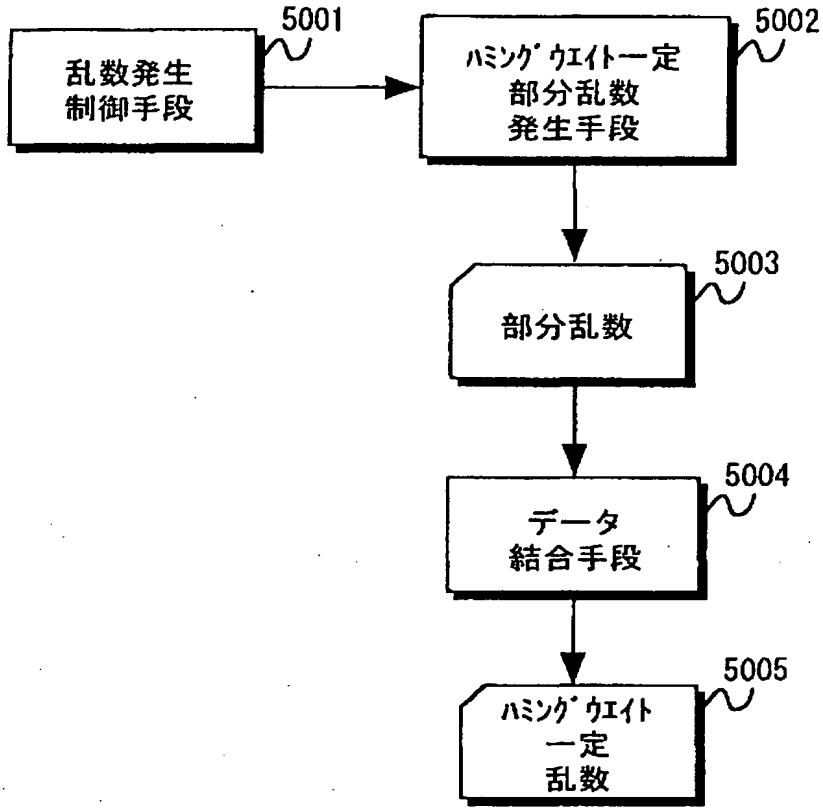
【図４９】

図　49


4901
乱数発生手段
乱数再生成制御信号
4902
乱数
4903
ﾊﾐﾝｸﾞｳｴｲﾄ
計算手段
4904
ﾊﾐﾝｸﾞｳｴｲﾄ
4905
ﾊﾐﾝｸﾞｳｴｲﾄ
検査手段

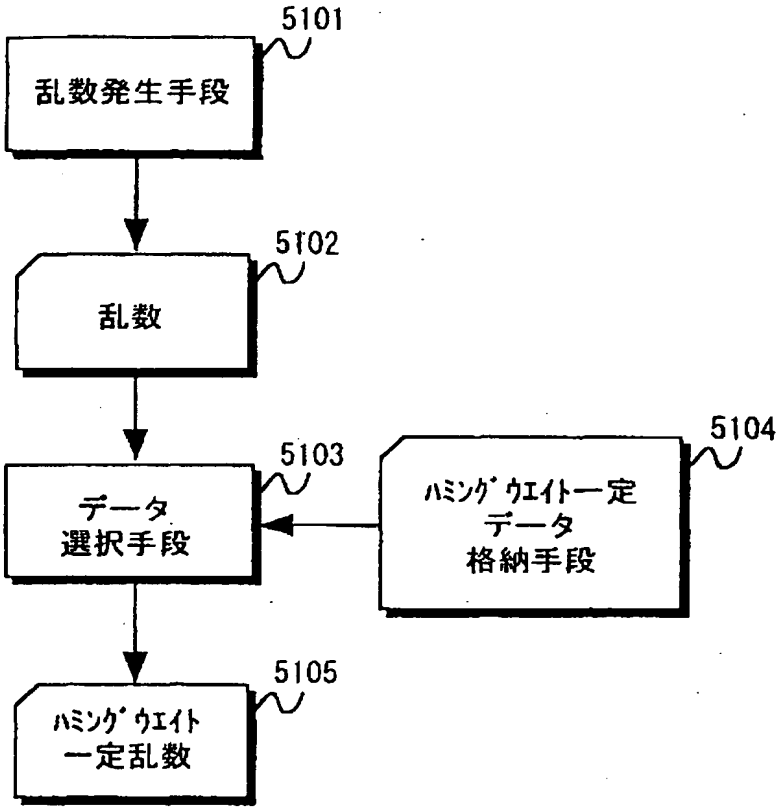
【図５０】

図　50


5001
乱数発生
制御手段
5002
ﾊﾐﾝｸﾞｳｴｲﾄ一定
部分乱数
発生手段
5003
部分乱数
5004
データ
結合手段
5005
ﾊﾐﾝｸﾞｳｴｲﾄ
一定
乱数

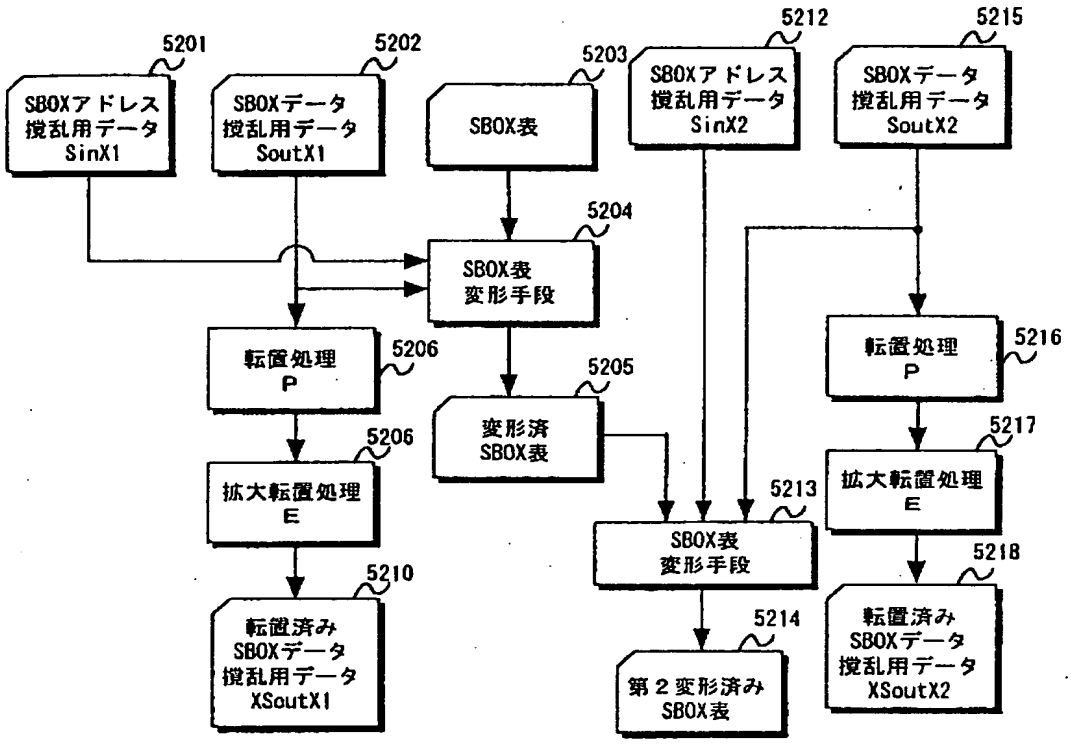
【図５１】

図 51


5101
乱数発生手段
5102
乱数
5103
データ
選択手段
5104
ﾊﾐﾝｸﾞｳｴｲﾄ一定
データ
格納手段
5105
ﾊﾐﾝｸﾞｳｴｲﾄ
一定乱数

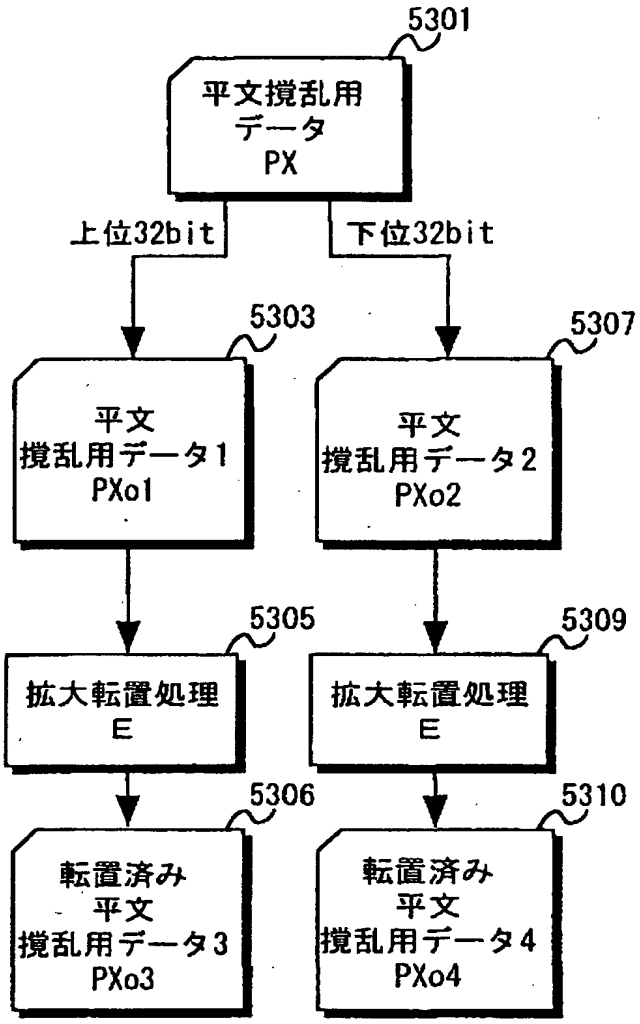
【図５２】

図 52


5201
SBOXアドレス
撹乱用データ
SinX1
5202
SBOXデータ
撹乱用データ
SoutX1
5203
SBOX表
5212
SBOXアドレス
撹乱用データ
SinX2
5215
SBOXデータ
撹乱用データ
SoutX2
5204
SBOX表
変形手段
転置処理
P
5206
5205
変形済
SBOX表
転置処理
P
5216
5206
拡大転置処理
E
5217
拡大転置処理
E
5213
SBOX表
変形手段
5218
5210
転置済み
SBOXデータ
撹乱用データ
XSoutX1
5214
第２変形済み
SBOX表
転置済み
SBOXデータ
撹乱用データ
XSoutX2

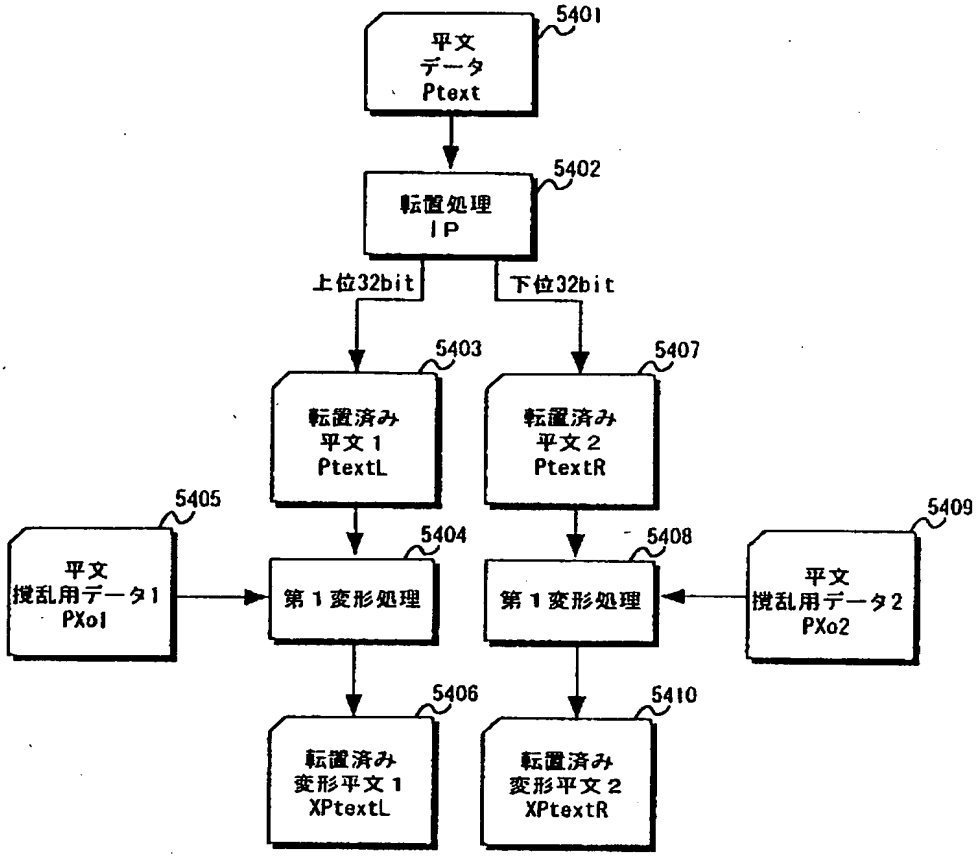
【図５３】

図　53


5301
平文撹乱用
データ
PX
上位32bit
下位32bit
5303
5307
平文
撹乱用データ1
PXo1
平文
撹乱用データ2
PXo2
5305
5309
拡大転置処理
E
拡大転置処理
E
5306
5310
転置済み
平文
撹乱用データ3
PXo3
転置済み
平文
撹乱用データ4
PXo4

【図５４】

図　54



【図５５】

図　55

（データ処理が左ローテートの場合）

1201

X1i	X1o
0x55555555	0xaaaaaaaa
0x33333333	0x66666666
0x66666666	0xcccccccc

【図５６】

図　56

1501

第1撹乱用
0x1c71c71c71c7
0x38e38e38e38e
0x34d34d34d34d
0x2cb2cb2cb2cb
0x38e38e38e38e

1502

第2撹乱用
0x55555555
0xAAAAAAAA
0x0F0F0F0F
0x71717171
0x5a5a5a5a

【図５７】

図 57

第1撹乱用データ＝0x1c71c71c71c7,
第2撹乱用データ＝0x55555555　の場合

1507

Index	Value	Index	Value	Index	Value	Index	Value
0	0xEFA72C4D	16	0x3911803A	32	0x40DA4917	48	0xF5BFF7A0
1	0x03DDEAD1	17	0xAC2456EC	33	0xFD13B462	49	0x5B496B9F
2	0x410DC1B2	18	0xA7D25DC9	34	0x1E662E4B	50	0xC81190F6
3	0xFD78BF0F	19	0x60870135	35	0xC8AF83B1	51	0xB6F4FE5C
4	0xD89E4A28	20	0x62C83393	36	0xE7491FB4	52	0x9C23C46A
5	0x740B24BD	21	0xC152FD56	37	0x8AD0C2DE	53	0x37E50109
6	0x1EE31FE4	22	0xCD75F47E	38	0x8B90B5D1	54	0x76CE5A8D
7	0x4795C278	23	0xBAECAECB	39	0x21067C87	55	0xEC3B97F0
8	0x266079F6	24	0x5CBBDE55	40	0xDA8CA2C9	56	0x3955610F
9	0xEF36474A	25	0x96C13020	41	0x436A1914	57	0xA0BCA6E3
10	0xFB36A20F	26	0x904C07A0	42	0x64FBD83C	58	0xA3A23D53
11	0x224F7C93	27	0x59BA9BFE	43	0x9F91E54A	59	0x05574025
12	0xB3F9B68B	28	0x0524E56C	44	0x2D377C7E	60	0x52E80B95
13	0xD860D917	29	0x3BFE8389	45	0x148D2FA8	61	0x6E225836
14	0x845A68D1	30	0x7A8F9B17	46	0xB10D83E2	62	0x0F74E628
15	0x1EA315A4	31	0x85196862	47	0x7278DA7D	63	0xD9CE3DCB

1507

Index	Value	Index	Value	Index	Value	Index	Value
0	0x12C0972D	16	0xEFB9FB9E	32	0x745329D2	48	0xB96EC2A5
1	0x4BB64AB1	17	0x9820A12B	33	0xDEC5E084	49	0x239B0FD8
2	0x215E71E8	18	0x9407A803	34	0xDF85978B	50	0x62B0545C
3	0x8DCB1F7D	19	0x379D66C6	35	0xB21C4AE1	51	0xC976913F
4	0xA82DEA5A	20	0x35D25460	36	0x9DFAD6E4	52	0xE3A1AB09
5	0x145894E7	21	0xF287089C	37	0x4B337B1E	53	0x9D44C5A3
6	0x5688BF84	22	0xF97103B9	38	0xA846E137	54	0x0E1C3ECA
7	0xBAF27918	23	0x6C44D56F	39	0x158F1C42	55	0xA0EAA2F5
8	0x4BF640F1	24	0xD04C3D37	40	0x272D8F28	56	0x8C9B689E
9	0xD10F3D84	25	0x2FDACE42	41	0xE458D6B7	57	0x5A21B37D
10	0x8D358C42	26	0x6EABD6DC	42	0x41D87AFD	58	0x3B770D63
11	0xE6ACE3DE	27	0x5071B039	43	0x7862292B	59	0x07BD5EC0
12	0x771A29C6	28	0x0CEFCEAB	44	0xCAC4B01F	60	0x50021570
13	0xAE63F75A	29	0xC51952F5	45	0x31AE8D69	61	0xF6F76806
14	0xBA63121F	30	0xC3946575	46	0x163F4C41	62	0xF5E9F3B6
15	0x73352CA3	31	0x09EE8B00	47	0x8FD9F79C	63	0x6C00345A

【図５８】

図　58

1602

Index	Value	Index	Value	Index	Value	Index	Value	Index	Value
0	0x12C0972D	16	0xEFB9FB9E	32	0x745329D2	48	0xB96EC2A5	第1撹乱用	0x1c71c71c71c7
1	0x4BB64AB1	17	0x9820A12B	33	0xDEC5E084	49	0x239B0FD8	第2撹乱用	0x55555555
2	0x215E71E8	18	0x9407A803	34	0xDF85978B	50	0x62B0545C		
3	0x8DCB1F7D	19	0x379D66C6	35	0xB21C4AE1	51	0xC976913F		
4	0xA82DEA5A	20	0x35D25460	36	0x9DFAD6E4	52	0xE3A1AB09		
5	0x145894E7	21	0xF287089C	37	0x4B337B1E	53	0x9D44C5A3		
6	0x5688BF84	22	0xF97103B9	38	0xA846E137	54	0x0E1C3ECA		
7	0xBAF27918	23	0x6C44D56F	39	0x158F1C42	55	0xA0EAA2F5		
8	0x4BF640F1	24	0xD04C3D37	40	0x272D8F28	56	0x8C9B689E		
9	0xD10F3D84	25	0x2FDACE42	41	0xE458D6B7	57	0x5A21B37D		
10	0x8D358C42	26	0x6EABD6DC	42	0x41D87AFD	58	0x3B770D63		
11	0xE6ACE3DE	27	0x5071B039	43	0x7862292B	59	0x07BD5EC0		
12	0x771A29C6	28	0x0CEFCEAB	44	0xCAC4B01F	60	0x50021570		
13	0xAE63F75A	29	0xC51952F5	45	0x31AE8D69	61	0xF6F76806		
14	0xBA63121F	30	0xC3946575	46	0x163F4C41	62	0xF5E9F3B6		
15	0x73352CA3	31	0x09EE8B00	47	0x8FD9F79C	63	0x6C0D345A		

【書類名】要約書

【要約】

【課題】　本願発明は、高いセキュリティを持つカード部材などの耐タンパー情報処理装置を提供するものである。技術的な課題は、カード部材、例えばＩＣカード用チップでのデータ処理と消費電流との関連性を減らすことである。

【解決手段】　本願発明の着眼点は、ＩＣカード用チップで消費される電流値と、処理されているデータの関連性を減らすための方法として、処理するデータを撹乱用データで変形し、データの処理を変形したデータで処理し、処理後に撹乱用データを用いて逆変換し、正しい処理結果を求めるものである。

【選択図】図４

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　　東京都千代田区神田駿河台４丁目６番地
氏　名　　株式会社日立製作所